

# Estimation de densités complexes par permutation de sous-échantillons

S. Dubuisson

C. Gonzales

Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie  
4 place Jussieu, 75005 Paris, FRANCE  
nom.prenom@lip6.fr

## Résumé

*Dans cet article, nous proposons une approche originale d'estimation séquentielle de densités non paramétriques définies dans des espaces de grande dimension, dans le cadre méthodologique du filtrage particulaire. En exploitant les indépendances conditionnelles de l'espace d'état, nous proposons de permuter des sous-ensembles indépendants de particules de manière à générer un nouvel ensemble échantillonnant mieux cet espace. Nous intégrons cette approche dans deux versions classiques du filtre particulaire : celui avec échantillonnage partitionné et celui à recuit simulé de manière à prouver son efficacité. Nous nous comparons aux approches classiques dans le cadre de l'estimation des densités d'objets synthétiques articulés. Nous montrons que notre approche diminue à la fois les erreurs d'estimation et les temps de traitement.*

## Mots Clef

Approximation de densité, filtrage particulaire, réseaux bayésiens.

## Abstract

*In this paper, we propose an original approach for sequential non-parametric density estimation defined in high-dimensional state spaces using the particle filtering framework. By exploiting conditional independences in the state space, we propose to swap independent particle subsets to generate new sets that better sample this space. We integrate this approach into two versions of particle filter, i.e., partition sampling and annealed particle filter, to prove its efficiency. We compare it to classical approaches on synthetic articulated object density estimation problems, and show that our approach reduces both estimation errors and computation times.*

## Keywords

Density approximation, particle filtering, Bayesian networks.

## 1 Introduction

En Intelligence Artificielle comme en Vision, il est très important aujourd'hui d'être capable de traiter des problèmes définis sur des espaces d'état et d'observation de

grande dimension. En Vision notamment, l'augmentation de la qualité des données à traiter et leurs multiples provenances induisent ce type de situation. Dans le cadre de cet article, nous nous intéressons à l'estimation de densités non paramétriques de grande dimension et évoluant au cours du temps, et avons choisi de nous positionner dans le cadre méthodologique des méthodes séquentielles de Monte Carlo (filtre particulaire). Il a été montré dans [9] que le nombre de particules nécessaires pour une bonne estimation de la densité augmente de manière exponentielle avec la dimension de l'espace d'état, ce qui rend ces méthodes séquentielles inutilisables en temps réel.

Dans cet article, nous proposons d'exploiter les indépendances conditionnelles dans l'espace d'état en transformant à l'aide d'une procédure de permutation des ensembles de particules pondérées en de nouveaux ensembles qui échantillonnent mieux la densité, et en particulier ses modes. Cette approche nous permet ainsi de gérer les grands espaces d'état, d'améliorer la qualité d'estimation tout en réduisant le nombre de particules nécessaires et donc les temps de calcul.

Cet article est organisé de la manière suivante. La section 2 donne des rappels sur le filtrage particulaire, les notations qui seront utilisées, ainsi qu'un bref état de l'art sur la prise en compte des grands espaces. La section 3 détaille l'approche proposée. La section 4 donne les résultats expérimentaux en termes de qualité d'estimation et de temps de calcul. Enfin, nous concluons et donnons quelques perspectives dans la section 5.

## 2 Filtrage particulaire (PF)

### 2.1 Contexte théorique

Dans ce contexte méthodologique [6], on cherche à estimer une séquence d'états  $\{\mathbf{x}_t\}_{t=1,\dots,T}$  dont l'évolution est donnée par l'équation  $\mathbf{x}_{t+1} = \mathbf{f}_{t+1}(\mathbf{x}_t, \mathbf{n}_{t+1}^x)$ , à partir d'un ensemble d'observations  $\{\mathbf{y}_t\}_{t=1,\dots,T}$ , reliées à l'état par l'équation  $\mathbf{y}_{t+1} = \mathbf{h}_{t+1}(\mathbf{x}_{t+1}, \mathbf{n}_{t+1}^y)$ . Classiquement,  $\mathbf{f}_{t+1}$  et  $\mathbf{h}_{t+1}$  sont des fonctions non linéaires variant dans le temps, et  $\mathbf{n}_{t+1}^x$  et  $\mathbf{n}_{t+1}^y$  des séquences de bruits i.i.d. Ces équations amènent naturellement à représenter ce type de problème à l'aide de la chaîne de Markov de la figure 1. Si l'on considère les équations ci-dessus d'un point de vue probabiliste, on peut alors

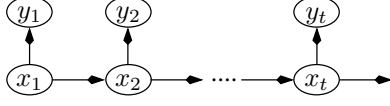


FIGURE 1 – Chaîne de Markov pour l'estimation d'états.

**Input** : ensemble de particules  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$ , image  $\mathcal{I}$

**Output** : ensemble de particules  $\{\mathbf{x}_{t+1}^{(i)}, w_{t+1}^{(i)}\}$

$\{\hat{\mathbf{x}}_{t+1}^{(i)}\} \leftarrow$  propagation ( $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$ )

$\{\hat{w}_{t+1}^{(i)}\} \leftarrow$  correction ( $\{\hat{\mathbf{x}}_{t+1}^{(i)}, w_t^{(i)}\}, \mathcal{I}$ )

$\{\mathbf{x}_{t+1}^{(i)}, w_{t+1}^{(i)}\} \leftarrow$  rééchantillonnage ( $\{\hat{\mathbf{x}}_{t+1}^{(i)}, \hat{w}_{t+1}^{(i)}\}$ )

**return**  $\{\mathbf{x}_{t+1}^{(i)}, w_{t+1}^{(i)}\}$

ALGORITHME 1 : Filtre particulaire PF

les calculer en deux étapes. D'abord, la prédiction de la densité  $p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \int_{\mathbf{x}_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t$ , avec  $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$  la densité *a priori* correspondant à la fonction de transition  $\mathbf{f}_{t+1}$ , puis une étape de correction  $p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) \propto p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})$ , avec  $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})$  la vraisemblance, correspondant à la fonction d'observation  $\mathbf{h}_{t+1}$ .

Le filtrage particulaire (cf. algorithme 1) a pour but d'approcher ces distributions par un ensemble de  $N$  échantillons pondérés  $\{\mathbf{x}_{t+1}^{(i)}, w_{t+1}^{(i)}\}$ , ou *particules*, qui correspondent à des réalisations possibles de l'état. Là encore, il se divise en deux étapes : (i) une étape de prédiction (utilisant les observations passées), qui consiste à propager l'ensemble de particules  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$  selon une fonction de proposition  $q(\mathbf{x}_{t+1}|\mathbf{x}_{0:t}, \mathbf{y}_{t+1})$ , suivie par (ii) une correction de cette prédiction (à l'aide d'une nouvelle observation), consistant à pondérer les particules à l'aide d'une fonction de vraisemblance, de manière à avoir  $w_{t+1}^{(i)} \propto w_t^{(i)} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^{(i)}) \frac{p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_{0:t}^{(i)})}{q(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_{0:t}^{(i)}, \mathbf{y}_{t+1})}$ , avec  $\sum_{i=1}^N w_{t+1}^{(i)} = 1$ . Les particules peuvent alors être rééchantillonnées : celles avec les plus forts poids sont dupliquées, et celles avec les plus faibles sont éliminées. L'estimation de la densité *a posteriori* est alors donnée par  $\sum_{i=1}^N w_{t+1}^{(i)} \delta_{\mathbf{x}_{t+1}^{(i)}}(\mathbf{x}_{t+1})$ , où  $\delta_{\mathbf{x}_{t+1}^{(i)}}$  sont des masses de Dirac centrées sur les particules  $\mathbf{x}_{t+1}^{(i)}$ .

## 2.2 Gestion des grands espaces d'état

La manipulation des distributions de probabilité ci-dessus nécessite des espaces d'état et d'observation de petites dimensions. Lorsque ce n'est pas la cas, il existe essentiellement deux manières pour rendre le processus viable : soit en réduisant la dimension des espaces à parcourir, en faisant des approximations ou des recherches locales, soit en décomposant ces espaces en produits cartésiens de sous-espaces de tailles raisonnables et en exploitant les indépendances conditionnelles probabilistes entre ces derniers.

Pour la première famille d'approches, on citera en particulier les méthodes d'optimisation. Il en existe un grand

**Input** : particules  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$  de la couche  $l$ , image  $\mathcal{I}$

**Output** : particules  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$  de la couche  $l + 1$

$\{\tilde{w}_t^{(i)}\} \leftarrow$  changement d'échelle ( $\{w_t^{(i)}\}$ )

$\{\bar{\mathbf{x}}_t^{(i)}, \bar{w}_t^{(i)}\} \leftarrow$  rééchantillonnage ( $\{\mathbf{x}_t^{(i)}, \tilde{w}_t^{(i)}\}$ )

$\{\hat{\mathbf{x}}_t^{(i)}\} \leftarrow$  propagation ( $\{\bar{\mathbf{x}}_t^{(i)}, \bar{w}_t^{(i)}\}$ )

$\{\hat{w}_t^{(i)}\} \leftarrow$  correction ( $\{\hat{\mathbf{x}}_t^{(i)}, \bar{w}_t^{(i)}\}, \mathcal{I}$ )

**return**  $\{\hat{\mathbf{x}}_t^{(i)}, \hat{w}_t^{(i)}\}$

ALGORITHME 2 : Annealing

**Input** :  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$ , image  $\mathcal{I}$ , nombre de couches  $L$

**Output** : particules  $\{\mathbf{x}_{t+1}^{(i)}, w_{t+1}^{(i)}\}$  au temps  $t + 1$

$\{\mathbf{x}_{t+1,0}^{(i)}, w_{t+1,0}^{(i)}\} \leftarrow$  PF ( $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$ )

**for**  $l = 1$  **to**  $L$  **do**

$\{\mathbf{x}_{t+1,l}^{(i)}, w_{t+1,l}^{(i)}\} \leftarrow$  Annealing ( $\{\mathbf{x}_{t+1,l-1}^{(i)}, w_{t+1,l-1}^{(i)}\}$ )

**end**

**return**  $\{\mathbf{x}_{t+1,L}^{(i)}, w_{t+1,L}^{(i)}\}$

ALGORITHME 3 : Le filtre particulaire à recuit simulé APF.

nombre, telles que le *path relinking* [13], ou une descente de gradient améliorée [2]. Ici, nous nous focalisons sur le filtre particulaire à recuit simulé (*Annealed Particle Filter* – APF) [3], qui est probablement le plus utilisé et qui consiste à ajouter à l'étape de rééchantillonnage des itérations (ou couches) de recuit simulé, de manière à mieux diffuser les particules dans l'espace d'état (cf. algorithme 3).

Pour la seconde famille d'approches, citons le *Partitioned Sampling* (PS) [8], qui exploite la décomposition de la densité de vraisemblance ainsi que de la dynamique dans des espaces de dimension réduite. Ainsi, les espaces d'état  $\mathcal{X}$  et d'observation  $\mathcal{Y}$  sont décomposés de la manière suivante :  $\mathcal{X} = \mathcal{X}^1 \times \dots \times \mathcal{X}^P$  et  $\mathcal{Y} = \mathcal{Y}^1 \times \dots \times \mathcal{Y}^P$ . Par exemple, afin de suivre les mouvements d'une personne,  $\mathcal{X}$  se décompose naturellement comme  $\mathcal{X}^{\text{buste}} \times \mathcal{X}^{\text{bras gauche}} \times \mathcal{X}^{\text{bras droit}}$ . L'exploitation d'indépendances conditionnelles entre ces sous-espaces amène à appliquer l'algorithme PF non plus en une seule fois sur  $(\mathcal{X}, \mathcal{Y})$  mais séquentiellement sur chacun des  $(\mathcal{X}^i, \mathcal{Y}^i)$ , ce qui accélère considérablement les temps de calcul. En effet, si l'on connaît la position du buste de la personne à suivre, on peut raisonnablement penser qu'il y a indépendance entre la position du bras gauche et celle du bras droit. Cela suggère donc de décomposer la distribution jointe sur  $(\mathcal{X}, \mathcal{Y})$  et sur l'ensemble des pas de temps  $t$ , comme indiqué dans le réseau bayésien dynamique (DBN) de la figure 2, où le buste correspond à la variable  $\mathbf{x}_t^1$ , le bras gauche à  $\mathbf{x}_t^2$  et le bras droit à  $\mathbf{x}_t^3$  [10]. Le modèle d'indépendance des DBN (la  $d$ -séparation, [11]) justifie alors l'application séquentielle de PF sur chacun des sous-espaces  $(\mathcal{X}^i, \mathcal{Y}^i)$ ,  $i = 1, \dots, P$ . PS est décrit plus formellement dans l'algorithme 4, où la notation  $(\hat{\mathbf{x}}_t^{(i),j}, \mathbf{x}_t^{(i),-j})$  représente le fait que la  $i^{\text{ème}}$  particule a le même état que  $\hat{\mathbf{x}}_t^{(i)}$  sur la partie  $j$  et le même état que  $\mathbf{x}_t^{(i)}$  sur les autres parties. Le problème est qu'en mul-

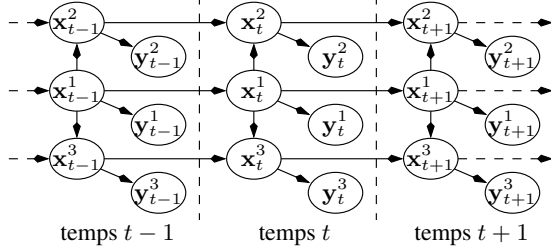


FIGURE 2 – Un réseau bayésien dynamique (DBN).

**Input** : particules  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$  au temps  $t$ , image  $\mathcal{I}$

**Output** : particules  $\{\mathbf{x}_{t+1}^{(i)}, w_{t+1}^{(i)}\}$  au temps  $t + 1$

$\{\mathbf{x}^{(i)}, w^{(i)}\} \leftarrow \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$

**for**  $j = 1$  **to**  $P$  **do**

$\{\hat{\mathbf{x}}^{(i),j}\} \leftarrow$  propagation ( $\{\mathbf{x}^{(i),j}\}$ )

$\{\hat{w}^{(i)}\} \leftarrow$  correction ( $\{(\hat{\mathbf{x}}^{(i),j}, \mathbf{x}^{(i),-j}), w^{(i)}\}, \mathcal{I}$ )

$\{\mathbf{x}^{(i)}, w^{(i)}\} \leftarrow$  rééch. ( $\{(\hat{\mathbf{x}}^{(i),j}, \mathbf{x}^{(i),-j}), \hat{w}^{(i)}\}$ )

**end**

**return**  $\{\mathbf{x}^{(i)}, w^{(i)}\}$

ALGORITHME 4 : Partitioned Sampling PS.

Multipliant les sous-espaces de traitement, on multiplie également le nombre d'étapes de rééchantillonnage, ce qui bruite énormément le processus d'estimation et dégrade les performances au cours du temps. Une idée similaire fondée sur les DBNs est exploitée dans [7] : la fonction de proposition  $q$  est décomposée comme le produit des densités conditionnelles en chaque nœud du réseau et PF est appliqué séquentiellement sur ces nœuds selon un ordre topologique dans le DBN. Ces deux approches sont combinées dans [12], pour définir l'algorithme du filtre particulière de manière à entièrement l'intégrer dans un DBN, ce qui peut être vu comme une généralisation de PS. D'autres méthodes fondées sur les réseaux bayésiens ont également été développées pour gérer les espaces de grandes dimensions. Par exemple, la technique non paramétrique de propagation de croyance (*Belief Propagation*) [14], ou encore le PF Rao-Blackwellisé pour DBN (RBPF) [4] qui décompose l'espace d'état en deux parties selon l'hypothèse forte que la distribution conditionnelle de la seconde partie peut être estimée par des méthodes classiques (filtre de Kalman par exemple). L'approche décrite dans [1] est probablement la plus proche de la notre. On y décrit un algorithme parallélisé de PF qui utilise la même décomposition de la probabilité jointe que dans les DBN afin de réduire le nombre de particules. L'espace d'état est divisé en sous-espaces dans lesquels les particules sont générées de manière indépendante à l'aide de différentes densités de proposition  $q$ . Cette approche offre une manière très flexible de choisir  $q$  pour échantillonner chaque sous-espace. En revanche, cela demande au DBN d'avoir une structure d'indépendance particulière, ce qui limite la généralisation de l'algorithme. Dans cet article, nous adressons des problèmes plus généraux, ce qui est décrit dans la section suivante.

### 3 Filtrage particulière avec permutation de sous-échantillons

Comme nous l'avons vu, l'idée clé de PS et de ses variations consiste à exploiter les indépendances probabilistes encodées par les DBN afin de ne réaliser les calculs du filtre particulière que dans des sous-espaces  $(\mathcal{X}^i, \mathcal{Y}^i)$  de petite dimension. Ces espaces étant "petits", les distributions de probabilités conditionnelles à estimer ont également beaucoup moins de paramètres que celle définie sur  $(\mathcal{X}, \mathcal{Y})$ , ce qui réduit exponentiellement le nombre de particules nécessaires à leur estimation et accélère donc d'autant les calculs. Cependant, on peut observer que PS n'exploite pas optimalement les indépendances du DBN. En effet, reprenons le DBN de la figure 2. Par  $d$ -séparation,  $\mathbf{x}_t^2$  et  $\mathbf{x}_t^3$  sont indépendants conditionnellement à  $(\mathbf{x}_t^1, \mathbf{x}_{t-1}^2, \mathbf{x}_{t-1}^3)$ , ce qui implique que, dans PS, les parties 2 et 3 peuvent être propagées/corrigées en parallèle plutôt que séquentiellement tout en garantissant que la distribution estimée est correcte. Le premier avantage de procéder ainsi est que l'on n'a besoin que d'un seul rééchantillonnage au lieu de 2 pour le filtrage des parties 2 et 3. Le deuxième avantage, et c'est l'objet principal de cet article, est que l'on va pouvoir permuer, parmi certains ensembles de particules, les sous-états des parties 2 et 3 de manière à se concentrer sur les modes de la distribution à estimer sans toutefois modifier la distribution estimée elle-même. Cette idée a été initiée dans [5] et nous la formalisons complètement ici.

Pour cela, nous devons considérer, non plus les particules  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$  au temps  $t$ , mais leurs trajectoires totales, soit  $\{\mathbf{x}_{1:t}^{(i)}, w_{1:t}^{(i)}\}$ . À un facteur de normalisation près, celles-ci échantillonnent la distribution  $p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t})$ . De la même manière que les états  $\mathbf{x}_t$  se décomposent en variables  $\mathbf{x}_t^j$ , les poids  $w_t$  se décomposent en  $w_t^j = p(\mathbf{y}_t^j | \mathbf{x}_t^j)$ , c'est-à-dire la vraisemblance de la partie  $j$ . Un poids  $w_t$  est alors défini par  $w_t \propto \prod_{j=1}^P w_t^j$ . Alors,  $\{\mathbf{x}_{1:t}^{(i),j}, w_{1:t}^{(i),j}\}$  échantillonne la distribution  $p(\mathbf{x}_{1:t}^j, \mathbf{y}_{1:t}^j | \mathbf{Pa}(\mathbf{x}_{1:t}^j))$ , où  $\mathbf{Pa}(\mathbf{x}_{1:t}^j)$  représente  $\bigcup_{s=1}^t \{\text{parents des nœuds } \mathbf{x}_s^j \text{ dans le DBN}\}$ . Et :

$$p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = \prod_{j=1}^P p(\mathbf{x}_{1:t}^j, \mathbf{y}_{1:t}^j | \mathbf{Pa}(\mathbf{x}_{1:t}^j)).$$

Par exemple, sur le DBN de la figure 2,  $\{\mathbf{x}_{1:t}^{(i),2}, w_{1:t}^{(i),2}\}$ , c'est-à-dire l'ensemble des parties 2 des particules sur toutes les tranches de temps, échantillonne  $p(\mathbf{x}_{1:t}^2, \mathbf{y}_{1:t}^2 | \mathbf{x}_{1:t}^1)$ , la distribution de toutes les variables sur la ligne du haut conditionnellement à celles du milieu. De même,  $\{\mathbf{x}_{1:t}^{(i),1}, w_{1:t}^{(i),1}\}$  et  $\{\mathbf{x}_{1:t}^{(i),3}, w_{1:t}^{(i),3}\}$  échantillonnent respectivement  $p(\mathbf{x}_{1:t}^1, \mathbf{y}_{1:t}^1)$  et  $p(\mathbf{x}_{1:t}^3, \mathbf{y}_{1:t}^3 | \mathbf{x}_{1:t}^1)$ , et la loi jointe se décompose en produit des 3 distributions :

$$p(\mathbf{x}_{1:t}^1, \mathbf{y}_{1:t}^1) \times p(\mathbf{x}_{1:t}^2, \mathbf{y}_{1:t}^2 | \mathbf{x}_{1:t}^1) \times p(\mathbf{x}_{1:t}^3, \mathbf{y}_{1:t}^3 | \mathbf{x}_{1:t}^1).$$

Or, par  $d$ -séparation, l'ensemble  $\{\mathbf{x}_{1:t}^2, \mathbf{y}_{1:t}^2\}$  est indépendant de  $\{\mathbf{x}_{1:t}^3, \mathbf{y}_{1:t}^3\}$  conditionnellement à  $\{\mathbf{x}_{1:t}^1\}$ . Donc, si l'on permute deux "sous-particules"  $\{\mathbf{x}_{1:t}^{(i),3}, w_{1:t}^{(i),3}\}$  et

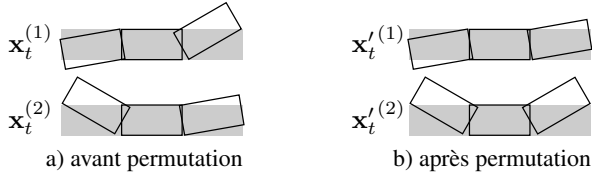


FIGURE 3 – L'opération de permutation.

$\{\mathbf{x}_{1:t}^{(j),3}, w_{1:t}^{(j),3}\}$  telles que  $\{\mathbf{x}_{1:t}^{(i),1}\} = \{\mathbf{x}_{1:t}^{(j),1}\}$ , cela n'aura d'impact ni sur l'estimation de  $p(\mathbf{x}_{1:t}^2, \mathbf{y}_{1:t}^2 | \mathbf{x}_{1:t}^1)$  (par indépendance) ni sur celle de  $p(\mathbf{x}_{1:t}^3, \mathbf{y}_{1:t}^3 | \mathbf{x}_{1:t}^1)$  (puisque permuter des éléments dans un échantillon n'affecte en rien l'estimation qu'il fournit). A fortiori, la distribution jointe estimée est inchangée. Cependant, cette opération permet de se focaliser sur les modes de cette distribution. En effet, considérons la figure 3, où 2 particules sont représentées,  $\mathbf{x}_t^1, \mathbf{x}_t^2, \mathbf{x}_t^3$  étant respectivement les parties centrale, gauche et droite de l'objet et les parties grisées représentant l'état réel de l'objet. Ces particules ont la même valeur pour  $\mathbf{x}_t^1$ , on peut donc permuter leurs valeurs de  $\mathbf{x}_t^3$ . Notons qu'avant permutation, la partie droite (resp. gauche) de la 1ère (resp. 2ème) particule était éloignée de l'état réel et induisait donc, globalement, un poids  $w_t$  peu élevé à la particule. Après permutation, la 1ère particule est très proche de l'état réel (et a donc un fort poids) et la deuxième en est assez éloignée (et a donc un très faible poids). Après rééchantillonnage, cette dernière sera donc supprimée et il ne restera que la 1ère particule, proche du mode de  $p$ .

Plus formellement, soit  $\mathcal{G}$  un DBN de variables aléatoires  $(\mathbf{x}_{1:t}^i, \mathbf{y}_{1:t}^i)$ ,  $i = 1, \dots, P$ . Comme usuellement, nous supposons que  $\mathcal{G}$  est un 2TBN, c'est-à-dire que  $\mathcal{G}$  ne contient que des arcs dont les extrémités appartiennent soit à la même tranche de temps soit à 2 tranches consécutives, et que le réseau est identique sur chaque tranche de temps. De plus, afin de simplifier la présentation de l'opération de permutation sans toutefois vraiment restreindre son champ d'application, nous supposons qu'à l'intérieur d'une tranche de temps, le réseau est une forêt (chaque nœud a au plus un parent dans la tranche) et que les seuls arcs entre les tranches sont du type  $(\mathbf{x}_{t-1}^j, \mathbf{x}_t^j)$ . Le DBN de la figure 2 vérifie ces hypothèses. Pour tout  $j \in \{1, \dots, P\}$ , appelons  $\text{Desc}(\{\mathbf{x}_t^j\})$  l'ensemble des descendants des variables de  $\{\mathbf{x}_t^j\}$  dans la tranche de temps  $t$ , et  $\text{Desc}(\{\mathbf{x}_{1:t}^j\}) = \cup_{s=1}^t \text{Desc}(\{\mathbf{x}_s^j\})$ . Soit  $\{P_1, \dots, P_K\}$  une partition de  $\{1, \dots, P\}$  telle que,  $\forall i, \forall j \in P_i$ , s'il existe  $\mathbf{x}_t^k \in \mathcal{G}$  tel que  $\mathbf{x}_t^j \in \text{Desc}(\{\mathbf{x}_t^k\})$ , alors  $k \in P_i$  avec  $h < i$ . Sur la figure 2,  $P_1 = \{\mathbf{x}_t^1\}$ ,  $P_2 = \{\mathbf{x}_t^2, \mathbf{x}_t^3\}$  est une telle partition. Chaque  $P_i$  correspond à un ensemble de parties que PS devrait propager/corriger en parallèle au temps  $t$ . Pour un  $P_i$  donné, l'opération de permutation, que nous noterons  $\rightleftharpoons^{P_i}$  est définie par : pour tout  $j \in P_i$ , soit  $\sigma_j$  une permutation des index des particules telle que seules des sous-particules ayant une même valeur de  $\mathbf{Pa}(\mathbf{x}_{1:t}^j)$  peuvent être permutées, alors toutes les sous-particules appartenant à  $\{\mathbf{x}_{1:t}^j\} \cup_{k \in \text{Desc}(\{\mathbf{x}_{1:t}^j\})} \{\mathbf{x}_{1:t}^k\}$  sont permutées selon  $\sigma_j$ .

**Input** :  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$ , partition  $\{P_1, \dots, P_K\}$ , image  $\mathcal{I}$

**Output** : particules  $\{\mathbf{x}_{t+1}^{(i)}, w_{t+1}^{(i)}\}$  au temps  $t + 1$

$\{\mathbf{x}_t^{(i)}, w_t^{(i)}\} \leftarrow \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$

**for**  $j = 1$  **to**  $K$  **do**

**for**  $k = 1$  **to**  $|P_j|$  **do**

$\{\mathbf{x}^{(i),k}\} \leftarrow \text{propagation}(\{\mathbf{x}^{(i),k}\})$

$\{w^{(i),k}\} \leftarrow \text{correction}(\{\mathbf{x}^{(i),k}\}, \mathcal{I})$

**end**

$\{\bar{\mathbf{x}}^{(i)}, \bar{w}^{(i)}\} \leftarrow \rightleftharpoons^{P_i}(\{\mathbf{x}^{(i)}, w^{(i)}\})$

$\{\mathbf{x}^{(i)}, w^{(i)}\} \leftarrow \text{rééchantillonnage}(\{\bar{\mathbf{x}}^{(i)}, \bar{w}^{(i)}\})$

**end**

**return**  $\{\mathbf{x}^{(i)}, w^{(i)}\}$

ALGORITHME 5 : Filtre avec permutation  $\overline{\overline{PS}}$ .

**Proposition 1** En propageant / corrigeant / permutant séquentiellement les sous-particules  $P_i$ ,  $i = 1, \dots, K$ , on échantillonne la distribution  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$ .

**Ébauche de démonstration:** Par définition, les  $P_i$  satisfont les hypothèses d'indépendance de PS et propager/corriger séquentiellement leurs particules conduit à échantillonner  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$ . Par  $d$ -séparation,  $\forall P_i, \forall j \in P_i$ ,  $\{\mathbf{x}_{1:t}^j\}$  est indépendant de  $\cup_{k \in P_i \setminus \{j\}} \{\mathbf{x}_{1:t}^k\}$  conditionnellement à  $\mathbf{Pa}(\mathbf{x}_{1:t}^j)$ . Donc, les sous-particules de chaque  $P_i$  peuvent être propagées/corrigées en parallèle sans modifier l'estimation de  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$ . De plus,  $\{\mathbf{x}_{1:t}^j\} \cup_{k \in \text{Desc}(\{\mathbf{x}_{1:t}^j\})} \{\mathbf{x}_{1:t}^k\}$  est indépendant du reste de  $\mathcal{G}$  conditionnellement à  $\mathbf{Pa}(\mathbf{x}_{1:t}^j)$  et échantillonne la distribution  $p(\mathbf{x}_{1:t}^{j \cup \text{Desc}(\{\mathbf{x}_{1:t}^j\})}, \mathbf{y}_{1:t}^{j \cup \text{Desc}(\{\mathbf{x}_{1:t}^j\})} | \mathbf{Pa}(\mathbf{x}_{1:t}^j))$ . Donc, effectuer une permutation d'un sous-ensemble de ces sous-particules dont la valeur de  $\mathbf{Pa}(\mathbf{x}_{1:t}^j)$  est identique ne peut altérer l'estimation de la distribution jointe.  $\square$

Comme on travaille sur des espaces continus, la probabilité que 2 sous-particules quelconques aient une même valeur de  $\mathbf{Pa}(\mathbf{x}_t^j)$  est extrêmement faible. En pratique, cela n'arrive que lorsqu'elles ont été engendrées par rééchantillonnage à partir d'une même particule. Dans ce cas, elles ont également une même valeur de  $\mathbf{Pa}(\mathbf{x}_{1:t}^j)$ . On s'autorise donc ici à permuter des particules dès lors qu'elles ont une même valeur de  $\mathbf{Pa}(\mathbf{x}_t^j)$ . L'avantage est que l'on n'a plus besoin de considérer les trajectoires des particules du temps 1 à  $t$ , mais seulement sur la tranche de temps  $t$ , ce qui conduit à l'algorithme 5. APF peut également être étendu afin d'intégrer l'opération de permutation. Pour cela, différentes possibilités existent, mais celle qui semble la plus efficace consiste à rajouter après chaque  $\rightleftharpoons^{P_i}$  de  $\overline{\overline{PS}}$  des couches d'anéaling portant sur  $P_i$  suivies, chacune, par une permutation  $\rightleftharpoons^{P_i}$ . Cet algorithme est appelé  $\overline{\overline{APF}}$ . Évidemment, certaines permutations sont meilleures que d'autres pour se focaliser sur les modes de la distribution. Une "bonne" permutation peut être obtenue assez simplement : d'après l'argument ci-dessus, si, pour  $j \in P_i$ , deux particules ont une même valeur de  $\mathbf{Pa}(\mathbf{x}_t^j)$ , elles ont probablement été engendrées par rééchantillonnage à partir

d'une même particule. Donc, pour tout  $k \in P_i$ , elles ont également une même valeur de  $\mathbf{Pa}(\mathbf{x}_t^k)$ . On peut donc partitionner l'ensemble  $N$  de particules en des sous-ensembles  $N_1, \dots, N_R$  tels que les particules d'un ensemble  $N_r$ ,  $r = 1, \dots, R$ , ont la même valeur de  $\mathbf{Pa}(\mathbf{x}_t^j)$  pour un  $j \in P_i$ . Pour chaque  $N_r$ , toutes les permutations possibles sont admissibles. Appelons  $\{r_1, \dots, r_s\}$  les éléments de  $N_r$ . Pour chaque  $j \in P_i$ , soit  $\sigma_j$  la permutation correspondant au tri des poids  $w_t^{(r_h).j}$ ,  $h = 1, \dots, s$ , par ordre décroissant. En appliquant  $\sigma_j$  pour tout  $j \in P_i$ , on obtient une opération de permutation  $\Rightarrow^{P_i}$  associant à la première particule l'ensemble meilleurs poids, à la seconde l'ensemble des seconds meilleurs poids, etc. Ainsi les premières particules sont-elles de poids très élevés et les dernières de poids très faibles (elles seront supprimées au prochain rééchantillonnage). La complexité en temps d'un tel algorithme pour l'ensemble des  $P_i$  est en  $O(PN(E + \log N))$ , où  $E$  représente la taille des variables  $\mathbf{x}_t^j$ . En effet, pour un  $P_i$  donné, en utilisant une table de hachage, la détermination des  $N_r$  peut être effectuée en  $O(N)$ . Pour chaque  $N_r$ , il y a  $|P_i|$  listes à trier, ce qui peut être réalisé, globalement en  $|P_i|N \log N$ . Enfin, l'application de la permutation modifie au plus  $P|\mathbf{x}_t^j|$  par particule et se fait donc, globalement, en  $O(NPE)$ .

## 4 Résultats expérimentaux

Nous avons choisi de focaliser notre étude sur la qualité de l'estimation séquentielle d'une densité, sans nous contraindre à des problèmes purement liés à l'image dans un premier temps. Pour cette raison, nous testons cette approche et la comparons à d'autres sur des séquences vidéo synthétiques que nous avons nous-mêmes générées, et qui contiennent toutes 300 images. Si elles sont visuellement très simples, elles contiennent de réelles difficultés pour un estimateur séquentiel, comme nous le verrons par la suite. Les objets dont nous cherchons à estimer la densité sont articulés et se déplacent et se déforment de manière aléatoire : on les modélise par un ensemble  $P$  de parties (ou régions), de formes de géométrie variable, dont les coins sont étiquetés  $C_i$ . Les objets ont une partie centrale, dont part un ensemble de  $|P_i|$  "bras",  $i = 1, \dots, K$  avec  $K$  la longueur d'un bras (voir section 3). Le vecteur d'état contient les paramètres décrivant chaque partie, et est défini par  $\mathbf{x}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \dots, \mathbf{x}_t^P\}$ , avec  $\mathbf{x}_t^i = \{x_t^i, y_t^i, \theta_t^i\}$ , où  $(x_t^i, y_t^i)$  est le centre de la partie  $i$ , et  $\theta_t^i$  son orientation,  $i = 1, \dots, P$ . On a donc  $|\mathcal{X}| = 3P$ . Une particule  $\mathbf{x}_t^{(j)} = \{\mathbf{x}_t^{(j),1}, \mathbf{x}_t^{(j),2}, \dots, \mathbf{x}_t^{(j),P}\}$ ,  $j = 1, \dots, N$ , est donc une configuration spatiale possible de l'objet articulé. Dans la première image, les particules sont générées uniformément autour de la position de l'objet. Pendant l'étape de prédiction elles sont propagées selon une marche aléatoire dont la variance a été empiriquement choisie. Les poids des particules sont calculés à partir de l'observation (image) courante, et sont donnés par  $w_{t+1}^{(j)} = w_t^{(j)} p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}^{(j)}) \propto w_t^{(j)} e^{-\lambda d^2}$ , avec  $\lambda = 50$  et  $d$  la distance de Bhattacharyya

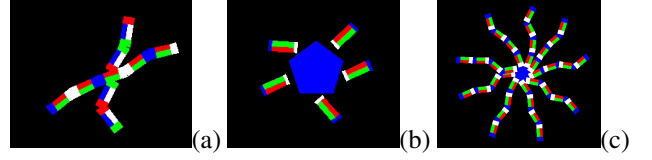


FIGURE 4 – Exemples d'objets (zoom) dont on cherche à estimer la densité de probabilité. De gauche à droite :  $|\mathcal{X}| = 27, 18$  et  $111$ .

entre les histogrammes cible (densité *a priori*) et référence (estimé à l'instant d'avant) de ces régions. La figure 4.(a) montre un objet avec  $P = 9$ ,  $|P_i| = 4$  et  $K = 2$ , soit  $|\mathcal{X}| = 27$ . Sur la figure 4.(b), nous avons un objet avec  $P = 6$ ,  $|P_i| = 5$ , et  $K = 1$ , soit  $|\mathcal{X}| = 18$ . Enfin, sur la figure 4.(c), on a  $P = 37$ ,  $|P_i| = 9$  et  $K = 4$  soit  $|\mathcal{X}| = 111$ .

L'estimation de la distribution d'un objet se fera toujours à partir de sa partie centrale. Selon les approches, on considérera simultanément les  $|P_i|$  parties indépendantes ou pas (voir la section 3). Les approches seront comparées essentiellement selon deux critères : le temps de traitement et l'erreur d'estimation. Cette erreur est donnée par la somme des distances euclidiennes entre chaque coin et son correspondant dans la vérité terrain. Nous comparerons les performances de PS sans et avec permutation, qui seront respectivement dénotés PS et  $\overline{\overline{PS}}$ , de l'APF classique, avec 1 ou 2 couches, et de l'APF avec permutation avec 1 ou 2 couches dénotés respectivement  $APF_1$  et  $APF_2$ ,  $\overline{\overline{APF_1}}$  et  $\overline{\overline{APF_2}}$ . L'ensemble des résultats présentés dans cet article sont une moyenne sur 20 lancers.

### 4.1 Erreur d'estimation

Nous considérons d'abord le cas d'objets avec  $|P_i| = 4$  (voir un exemple figure 4.(a)), et nous nous intéressons à l'erreur d'estimation en fonction du nombre total  $P$  de parties à traiter. Le nombre de particules est choisi tel que  $N = 10 \times P$ . Le tableau 1 reporte les résultats obtenus. Nous pouvons noter que, pour  $P$  fixé,  $\overline{\overline{PS}}$  donne des erreurs plus petites PS, pour des temps de traitement plus faibles, tout comme  $\overline{\overline{APF_1}}$  et  $\overline{\overline{APF_2}}$  sont toujours meilleurs que  $APF_1$  et  $APF_2$ . En particulier, on note que  $\overline{\overline{APF_1}}$  estime mieux que  $APF_2$ . Ainsi, notre approche permet à la fois de diminuer l'erreur et les temps de calcul. Un dernier aspect important concerne le fait que l'écart de temps de calcul entre les versions avec et sans permutation augmente avec  $P$ . On peut ainsi remarquer que pour  $P = 17$ ,  $\overline{\overline{PS}}$  est 15% plus rapide que PS, et  $\overline{\overline{APF_1}}$  est 4% plus rapide que  $APF_1$ . Cette diminution du temps de traitement est due au fait qu'en traitant simultanément  $|P_i|$  parties, nous diminuons le nombre de rééchantillonnages nécessaires. Cette étude sera l'objet de la section 4.3.

Les méthodes d'optimisation intégrées dans le PF per-

TABLE 1 – Erreurs d’estimation (pixels) et temps de calcul (sec.), pour un objet avec  $|P_i| = 4$ , en faisant varier  $P$ .

	PS	$\overline{\overline{PS}}$	APF <sub>1</sub>	APF <sub>2</sub>	$\overline{\overline{APF_1}}$	$\overline{\overline{APF_2}}$
P=5	226	196	190	170	160	154
	3.5	3.5	6.4	9.3	6.2	9.3
P=9	278	207	234	217	193	177
	11.2	10.2	20.9	31.1	19.9	30.4
P=13	693	567	549	516	499	478
	23.9	21.5	48.1	66.5	44	62.3
P=17	685	651	630	546	534	511
	46.4	39.1	87.6	125.3	84.2	120.9

TABLE 2 – Erreur d’estimation (pixels) et temps de traitement (sec.) de la densité de l’objet de la figure 5.

	PS	$\overline{\overline{PS}}$	APF <sub>1</sub>	APF <sub>2</sub>	$\overline{\overline{APF_1}}$	$\overline{\overline{APF_2}}$
$N$	120	120	40	40	40	40
Err.	818	627	799	581	496	493
Tps	23.9	23.9	21.2	30	20.9	30.5

mettent de diminuer le nombre de particules. Nous proposons de comparer les erreurs d’estimation de la densité d’un objet avec  $|P_i| = 3$  et  $P = 10$  ( $|\mathcal{X}| = 30$ ). Nous avons comparé PS et  $\overline{\overline{PS}}$  avec  $N = 120$ , et APF<sub>1</sub>, APF<sub>2</sub>,  $\overline{\overline{APF_1}}$  et  $\overline{\overline{APF_2}}$  avec  $N = 40$ . Pour plus de lisibilité, nous avons reporté les courbes d’erreurs d’estimation de PS, APF<sub>2</sub> et  $\overline{\overline{APF_1}}$  sur la figure 5.(a). On remarque que  $\overline{\overline{APF_1}}$  est le plus stable des trois approches, alors qu’il n’utilise qu’une couche. Un résultat de suivi est donné sur la figure 5.(b) (le modèle estimé est dessiné en trait jaune sur l’image). Le tableau 2 reporte les erreurs moyennes de suivi et les temps de calcul pour toutes les méthodes. On remarque que  $\overline{\overline{APF_1}}$  estime mieux que APF<sub>1</sub> et APF<sub>2</sub>, et ce en moins de temps de calcul que PS. Enfin, notre approche fait converger APF après une seule couche de recuit simulé, ce qui n’est pas le cas de la version sans permutation.

Un des principaux avantages de notre approche est sa capacité à utiliser les indépendances dans les espaces probabilistes pour décomposer les calculs et ainsi les accélérer. Nous testons donc l’influence de  $|P_i|$  sur l’erreur d’estimation : les résultats sont reportés dans le tableau 3, pour  $N = 50$  et  $|\mathcal{X}| = 3(|P_i|+1)$ . Avec ce nombre faible de particules, les estimations se dégradent considérablement avec l’augmentation de la taille de l’espace d’état traité (voir  $|P_i| = 8, 9, 10$ ). Toutefois, l’erreur d’estimation augmente moins fortement pour les méthodes avec permutation que pour celles sans.

## 4.2 Tests de convergence

Dans la section 4.1, nous ne nous sommes pas posé la question du nombre optimal  $N$  de particules. Nous avons comparé, en fonction de  $N$ , les qualités d’estimation de la densité d’un objet avec  $P = 9$  et  $|P_i| = 2$  ( $|\mathcal{X}| = 27$ ), voir figure 4.(a)). Les courbes de convergence sont données dans

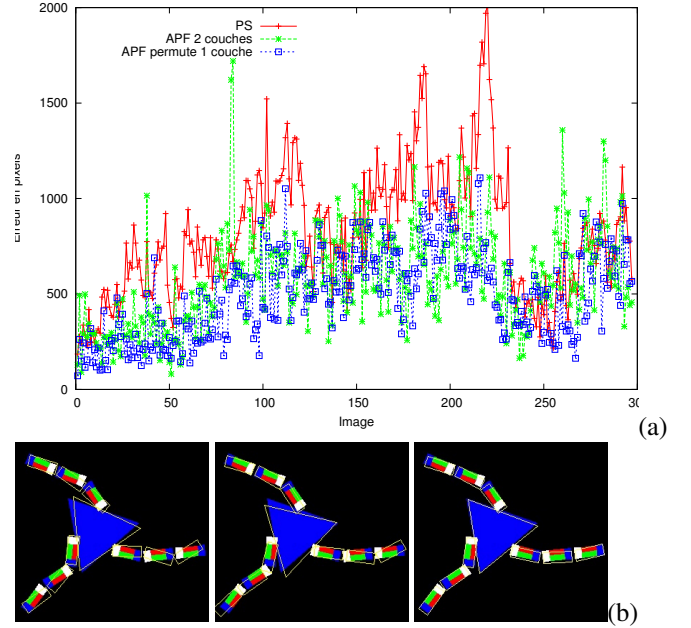


FIGURE 5 – Estimation de la densité d’un objet avec  $|P_i| = 3$  et  $P = 10$  ( $|\mathcal{X}| = 30$ ). Comparaison entre PS avec  $N = 120$  particules, APF<sub>2</sub> et  $\overline{\overline{APF_1}}$  avec  $N = 40$ . Haut : courbes d’erreur d’estimation image par image. Bas : un résultat de suivi, de gauche à droite, PS, APF<sub>2</sub>,  $\overline{\overline{APF_1}}$ . L’objet estimé est surimposé en trait jaune.

la figure 6, selon la quantité de bruit Gaussien présent dans les images de la séquence traitée ( $\sigma = \{0, 20\}$ ). Nous pouvons voir que la permutation permet de diminuer significativement les erreurs d’estimation (l’échelle des ordonnées est logarithmique) ainsi que la rapidité de convergence. En particulier, on peut voir que  $\overline{\overline{APF_1}}$  et  $\overline{\overline{APF_2}}$  donnent des erreurs plus petites et convergent plus vite que l’APF classique. Nous pouvons également constater que  $\overline{\overline{PS}}$  est quasi-équivalent à APF<sub>1</sub> : cela montre que notre approche est autant, sinon plus, robuste qu’une méthode d’optimisation telle que celle présente dans APF (le recuit simulé). Cela s’accroît avec l’augmentation du bruit dans la séquence :  $\overline{\overline{PS}}$  devient meilleur que APF et converge tout aussi rapidement.

Enfin, le tableau 4 donne les erreurs d’estimation et les temps de calcul en fonction du nombre de couches utilisées dans APF et  $\overline{\overline{APF}}$  pour  $N = 20$  et  $|\mathcal{X}| = 30$ . Là encore, notre approche reste plus robuste.

## 4.3 Temps de calcul

Il y a principalement quatre opérations qui interviennent dans le temps de calcul global : le temps de propagation des particules et de calcul de la vraisemblance pour l’étape de correction, qui sont similaires pour toutes les approches, et les temps de rééchantillonnage (incluant la normalisation) et de permutation qui sont spécifiques à chaque ap-



TABLE 3 – Erreurs (pixels) d’estimation de la densité en fonction de  $|P_i|$ , pour  $N = 50$  et  $|\mathcal{X}| = 3(|P_i| + 1)$ .

	PS	$\overline{\overline{PS}}$	APF <sub>1</sub>	APF <sub>2</sub>	$\overline{\overline{APF_1}}$	$\overline{\overline{APF_2}}$
$ P_i  = 2$	176	151	148	124	123	118
$ P_i  = 3$	180	170	152	140	126	119
$ P_i  = 4$	178	169	157	133	134	124
$ P_i  = 5$	341	293	262	271	238	214
$ P_i  = 6$	343	316	282	254	243	224
$ P_i  = 7$	467	417	432	391	379	342
$ P_i  = 8$	751	680	667	687	540	499
$ P_i  = 9$	488	417	422	376	353	332
$ P_i  = 10$	851	764	816	799	689	621

TABLE 4 – Erreur d’estimation de la densité (pixels) et temps de calcul (sec.) pour  $P = 10$ ,  $|P_i| = 2$  ( $|\mathcal{X}| = 30$ ). Variation du nombre  $L$  de couches pour APF et  $\overline{\overline{APF}}$ , avec  $N = 20$ .

	1	2	3	4	5	6	7
APF	707	594	516	532	551	542	544
	5.6	8.6	11.2	14.2	17.8	19.9	22.1
$\overline{\overline{APF}}$	491	461	473	448	420	419	420
	5.6	8.6	11.4	14.3	17.7	20	22.4

proche. Pour que notre méthode soit intéressante, il faut que l’introduction de la permutation ne ralentisse pas les temps de calcul, donc que le temps de rééchantillonnage plus celui de permutation soit au plus équivalent au temps de rééchantillonnage des approches classiques (PS et APF). Nous avons mesuré ces temps en prenant en compte différents critères. Les figures 7.(a-b) montrent les temps, en fonction de  $N$ , pour l’estimation séquentielle de la densité d’un objet avec  $K = 5$  et  $P = 6$  ( $|\mathcal{X}| = 18$ ). On remarque que notre approche réduit bien les temps de rééchantillonnage (en rouge), et que l’ajout de la permutation (en bleu) n’entraîne pas un surcoût. De plus, avec l’augmentation de  $N$ , notre approche diminue les temps de calcul. Par exemple, pour  $N = 1000$ , les temps totaux de traitement pour PS,  $\overline{\overline{PS}}$ , APF<sub>1</sub> et  $\overline{\overline{APF_1}}$  sont respectivement de 163, 157, 349 et 329 secondes, soit un gain de 4 à 6%. La figure 7.(c) montre les résultats en faisant varier  $|P_i|$ , avec  $N = 200$  et  $|\mathcal{X}| = 3(|P_i| + 1)$ . On remarque que le temps de rééchantillonnage pour  $\overline{\overline{PS}}$  et  $\overline{\overline{APF_1}}$  diminue avec  $K$ , de même que le temps total de traitement. Par exemple, pour  $|P_i| = 10$  ( $|\mathcal{X}| = 33$ ), les temps totaux de traitement pour PS,  $\overline{\overline{PS}}$ , APF<sub>1</sub> et  $\overline{\overline{APF_1}}$  sont respectivement de 51, 50, 109 et 105 secondes, soit un gain de 2 à 4%. Enfin, la figure 7.(d) montre les résultats en faisant varier  $P$ , avec  $|P_i| = 5$  et  $N = 200$  (i.e. on fait varier  $K$ , et  $|\mathcal{X}| = 3(K|P_i| + 1)$ ). Là encore, on constate que les temps diminuent avec l’augmentation de la longueur du bras. Pour  $K = 6$  ( $|\mathcal{X}| = 93$ ), les temps totaux de traitement pour PS,  $\overline{\overline{PS}}$ , APF<sub>1</sub> et  $\overline{\overline{APF_1}}$  sont respectivement de 106, 94, 224 et 207 secondes, soit

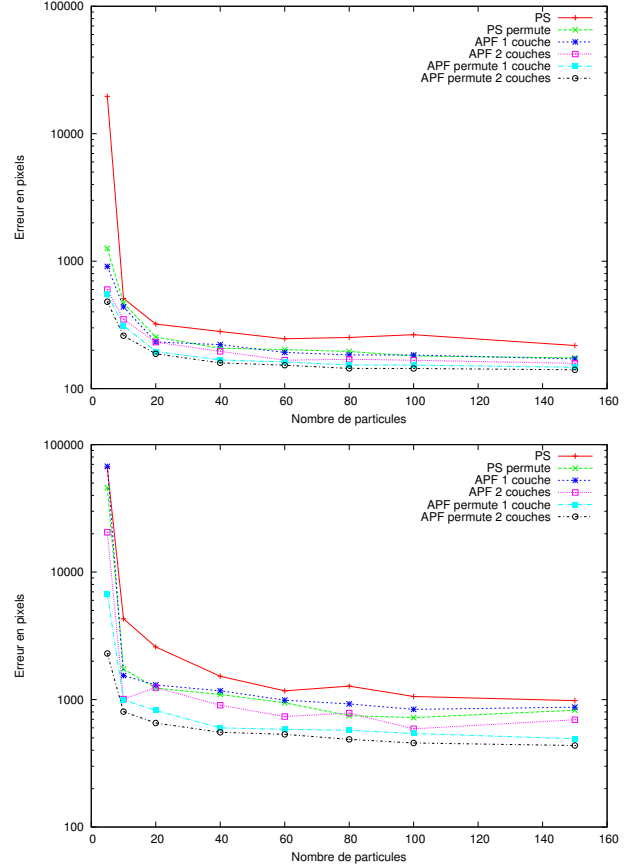


FIGURE 6 – Tests de convergence. Erreur d’estimation de la densité d’un objet pour  $P = 9$ ,  $|P_i| = 2$  (échelle logarithmique) en fonction de  $N$  pour PS,  $\overline{\overline{PS}}$ , APF<sub>1</sub>, APF<sub>2</sub>,  $\overline{\overline{APF_1}}$  et  $\overline{\overline{APF_2}}$ , selon la quantité de bruit Gaussien présent dans la séquence vidéo, de haut en bas :  $\sigma = 0$ ,  $\sigma = 20$ .

un gain de 7 à 11%.

## 5 Conclusion

Nous avons présenté dans cet article une nouvelle approche d’estimation séquentielle de densité qui vise (i) à exploiter les indépendances probabilistes encodées par les DBN afin de ne réaliser les calculs du filtre particulaire que dans des sous-espaces de petite dimension et (ii) faire des permutations de certains sous-ensembles de particules de manière à se concentrer sur les modes de la distribution à estimer. Nous avons proposé un cadre théorique rigoureux qui nous garantit que la distribution ainsi estimée n’est pas modifiée, ainsi qu’une analyse en complexité au pire cas. Les tests ont permis de montrer que notre approche, intégrant une étape supplémentaire par rapport aux autres (la permutation), n’est pas plus coûteuse. En effet, le nombre d’étapes de rééchantillonnage nécessaires est diminué avec le traitement simultané des parties indépendantes de la densité de l’objet à estimer. Ce gain de temps permet d’absorber le surcoût dû à l’étape de permutation, et ainsi de ne pas

augmenter les temps globaux de traitement. De plus, nous avons montré que plus la taille de l'espace traité augmente, plus notre approche permet de gagner du temps. Nos travaux actuels portent sur la définition d'un critère plus efficace pour la permutation des sous-ensembles de particules, en particulier à définir ce qu'est (ou pas) un bon nuage de particules.

## Références

- [1] E. Besada-Portas, S.M. Plis, J.M. Cruz, and T. Lane. Parallel subspace sampling for particle filtering in dynamic Bayesian networks. In *ECML PKDD*, pages 131–146, 2009.
- [2] M. Bray, E. Koller-Meier, P. Müller, N. N. Schraudolph, and L. Van Gool. Stochastic optimization for high-dimensional tracking in dense range maps. *IEEE Proceedings Vision, Image and Signal Processing*, 152(4) :501–512, 2005.
- [3] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61 :185–205, 2005.
- [4] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *UAI*, pages 176–183, 2000.
- [5] C. Gonzales, S. Dubuisson, and X.-S. Nguyen. Simultaneous partitioned sampling for articulated object tracking. In *ACIVS*, Lecture Notes in Computer Science, pages 150–161, 2011.
- [6] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings of Radar and Signal Processing*, 140(2) :107–113, 1993.
- [7] K. Kanazawa, D. Koller, and S. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *UAI*, pages 346–35, 1995.
- [8] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *ICCV*, pages 572–587, 1999.
- [9] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV*, pages 3–19, 2000.
- [10] K. Murphy. *Dynamic Bayesian Networks : Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002.
- [11] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufman Publishers, 1988.
- [12] C. Rose, J. Saboune, and F. Charpillat. Reducing particle filtering complexity for 3D motion capture using dynamic Bayesian networks. *AAAI*, pages 1396–1401, 2008.
- [13] A. Sánchez, J. Pantrigo, and K. Gianikellis. Combining Particle Filter and Population-based Metaheuristics for Visual Articulated Motion Tracking. *Electronic Letters on Computer Vision and Image Analysis*, 5(3) :68–83, 2005.
- [14] E.B. Sudderth, A.T. Ihler, M. Isard, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. *Communications of ACM*, 53 :95–103, 2010.

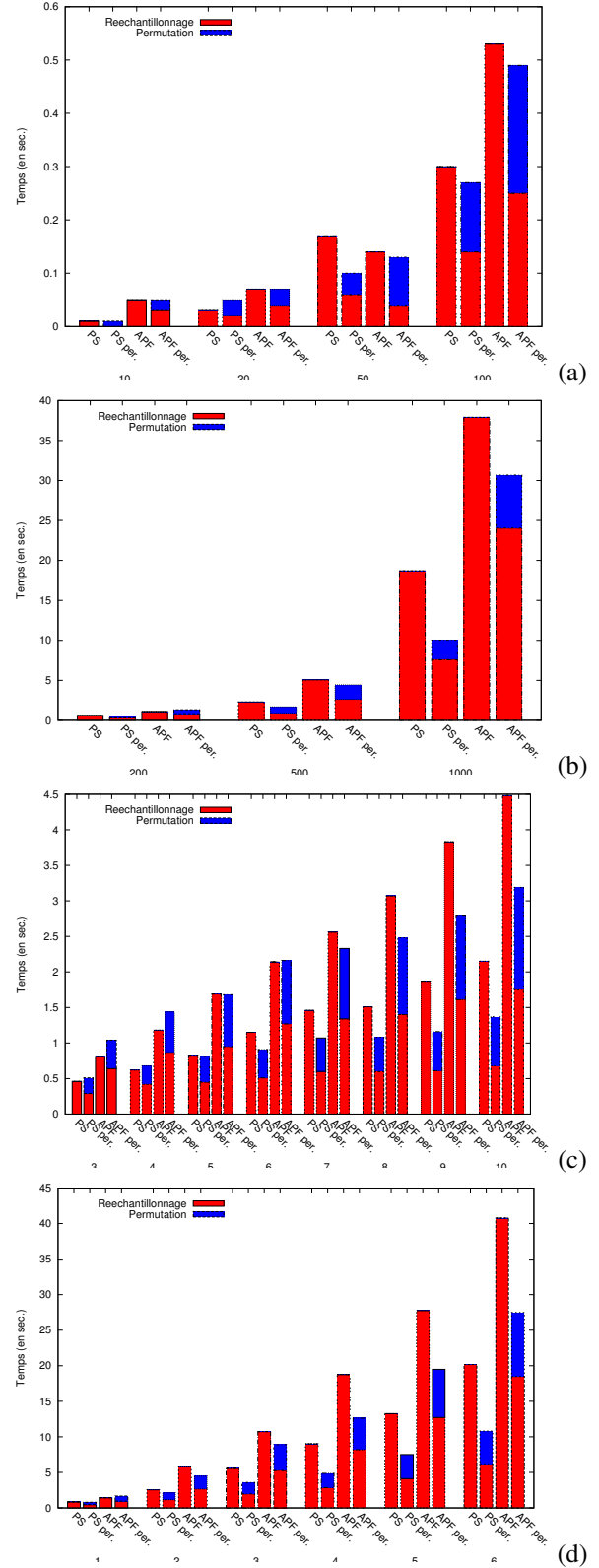


FIGURE 7 – Temps de rééchantillonnage (en rouge) et de permutation (en bleu) selon les approches (PS,  $\overline{PS}$ , APF<sub>1</sub>,  $\overline{APF_1}$ ). (a-b) En faisant varier  $N$  ( $|P_i| = 5, P = 6, |\mathcal{X}| = 18$ ). (b) En faisant varier  $|P_i|$  ( $N = 200, |\mathcal{X}| = 3(|P_i| + 1)$ ). (c) en faisant varier la longueur  $K$  des bras ( $|P_i| = 5, N = 200, |\mathcal{X}| = 3(K|P_i| + 1)$ ).