Les Cahiers du GERAD ISSN: 0711–2440

Filtering of Images for Detecting Multiple Targets Trajectories

I. Gentil, B. Rémillard
P. Del Moral

G–2002–72

December 2002

Les textes publiés dans la série des rapports de recherche HEC n'engagent que la responsabilité de leurs auteurs. La publication de ces rapports de recherche bénéficie d'une subvention du Fonds F.C.A.R.

Filtering of Images for Detecting Multiple Targets Trajectories

Ivan Gentil

Bruno Rémillard*

Service de l'enseignement des méthodes quantitatives de gestion
HEC Montréal
3000, chemin de la Côte Sainte-Catherine
Montréal (Québec) Canada H3T 2A7
** and GERAD*
{ivan.gentil,bruno.remillard}@hec.ca

Pierre Del Moral

Laboratoire de Statistique et Probabilités
Université Paul Sabatier
118 route de Narbonne
31062 TOULOUSE Cedex France
delmoral@cict.fr

December, 2002

Les Cahiers du GERAD

G–2002–72

Copyright © 2002 GERAD

Abstract

The aim of this paper is to present efficient algorithms for the detection of multiple targets in noisy images. The algorithms are based on the optimal filter of a multidimensional Markov chain signal. We also present some simulations, in the case of one, two and three targets, showing the efficiency of the method for detecting the positions of the targets.

Key words: Filtering, Markov chains, images, multiple targets.

Résumé

Le but de cet article est de présenter des algorithmes efficaces pour la recherche de cibles dans des images bruitées. Les algorithmes reposent sur le filtre optimal pour une chaîne de Markov multidimensionnelle représentant le signal. Nous présentons aussi des simulations, dans le cas d'une, deux et trois cibles, démontrant l'efficacité de la méthode proposée pour la détection de positions de cibles.

1 Introduction

The problem of detecting targets in a region, using images, is a quite interesting challenge, having a lot of applications. For example, one can try to find a boat lost at sea, or try to track drug smugglers, etc. Unfortunately, images are not crystal clear and can be affected by several parameters, in particular weather conditions.

In what follows, we will assume that the targets move on a lattice, according to a Markov chain. Moreover, the observations are black-and-white noisy images of a finite fixed region. Our setting is different of the setting considered by Ballantyne, Chan and Kouriztin (2002) [1], where they used different kind of noise and a different approach for the motion of targets.

In order to simplify calculations, the following notations will be used.

Model for targets: Let $(\xi_k)_{k\geq 0}$ be a finite system of m Markovian targets evolving in a countable space $\mathfrak{X}$. The state space is the set of all $\xi \in \{0,1\}^{\mathfrak{X}}$ so that $\displaystyle\sum_{x\in\mathfrak{X}} \xi(x) = m$.

Let $\mathcal{A}$ be the set of all $A \subset \mathfrak{X}$, $|A| = m$. For any $A \in \mathcal{A}$, define

$$F_A(\xi) = \prod_{x\in A} \xi(x),$$

and set $\nu_\xi(A) = E_P(F_A(\xi_{k+1})|\xi_k = \xi)$. Note that, for any $B \in \mathfrak{X}$, $|B| \leq m$,

$$E_P\left\{F_B(\xi_{k+1})|\xi_k = \xi\right\} = P(\xi_k(x) = 1, x \in B|\xi_k = \xi) = \sum_{A\in\mathcal{A}, A\supset B} \nu_\xi(A).$$

Further set

$$M(A,B) = E\left\{F_A(\xi_{k+1})|F_B(\xi_k) = 1\right\}, \tag{1}$$

where $A, B \in \mathcal{A}$. The transition function M is important since it describes exactly the movement of the m targets. We discuss about it in the section 3.

Model for observations: Observations are denoted by $Y_k \in \{0,1\}^I$, where $I \subset \mathfrak{X}$ is a finite set. Given $\xi_0, \ldots, \xi_k$, we assume the $\{Y_k(x)\}_{x\in I}$ are independent and

$$P(Y_k(x) = 0|\xi_k(x) = 0) = p_0, \tag{2}$$

$$P(Y_k(x) = 1|\xi_k(x) = 1) = p_1, \tag{3}$$

where $0 < p_0, p_1 < 1$.

Two algorithms will be defined to compute the optimal filter according as we know or we don't known the transition function M.

In the next section, we will present a straightforward algorithm when the motion of targets is assumed to be known, i.e. M is known. In section 3 we will propose an algorithm in the case on an unknown motion. Simulations showing the performance of the algorithms are presented in Section 4 in the case of $m = 1, 2, 3$ targets. Finally, section 5 contains a discussion of the results and ideas for future work.

2 Optimal filter when the transition mechanism is known

Throughout this section we assume the transition function M to be known. Our aim is to find an easy algorithm to compute

$$P\left(\cap_{x\in B}\{\xi_k(x)=1\}|\,\mathcal{Y}_k\right),$$

where $B \in \mathcal{A}$, $|B| \le m$ and $\mathcal{Y}_k$ is the sigma-algebra generated by observations $Y_1, \ldots, Y_k$. Set $\mathcal{Y}_0 = \{\emptyset, \{0,1\}^I\}$.

The first step is to compute, for any $y \in \{0,1\}^I$, the following conditional probability

$$P(Y_k = y|\xi_k = \xi) = P\left(\cap_{x\in I}\{Y_k(x) = y(x)\}|\,\xi_k = \xi\right).$$

Using the independence assumption, together with (2) and (3), one can check that

$$\begin{aligned} P(Y_k = y|\xi_k = \xi) &= \prod_{x\in I}\left\{(1-p_0)^{1-\xi(x)}p_1^{\xi(x)}\right\}^{y(x)} \\ &\quad \times \left\{p_0^{1-\xi(x)}(1-p_1)^{\xi(x)}\right\}^{1-y(x)} \\ &= p_0^{|I|}\left(\frac{1-p_0}{p_0}\right)^{<y>}\left(\frac{1-p_1}{p_0}\right)^{<\xi>}\left(\frac{p_0p_1}{(1-p_0)(1-p_1)}\right)^{<y\xi>} \\ &= \Lambda(y,\xi), \end{aligned} \tag{4}$$

where $< y >= \sum_{x\in I} y(x)$, $< \xi >= \sum_{x\in I} \xi(x)$, and $< y\xi >= \sum_{x\in I} y(x)\xi(x)$.

Let P be the joint law of the Markovian targets with initial distribution ν, and the observations, and let Q be the joint law of the Markovian targets with initial distribution ν, and independent Bernoulli observations with mean 1/2.

Further let $\mathcal{G}_k$ be the sigma-algebra generated by $Y_1, \ldots, Y_k, \xi_0, \ldots, \xi_k$. Then it is easy to check that with respect to $\mathcal{G}_k$, P is equivalent to Q and

$$\left.\frac{dP}{dQ}\right|_{\mathcal{G}_k} = 2^{|I|}\prod_{j=1}^{k}\Lambda(Y_j,\xi_j). \tag{5}$$

Further define

$$L_k = \prod_{j=1}^{k}\Lambda(Y_j,\xi_j). \tag{6}$$

It follows that for any $\mathcal{G}_k$-measurable random variable Z and for any sigma-algebra $\mathcal{F} \subset \mathcal{G}_k$,

$$E_P\left(Z|\mathcal{F}\right) = \frac{E_Q\left(ZL_k|\mathcal{F}\right)}{E_Q\left(L_k|\mathcal{F}\right)}. \tag{7}$$

While this formula is an easy consequence of the properties of conditional expectations, in the context of filtering, (7) is known as the Kallianpur-Stribel formula.

The key observation here is to note that expectations relative to Q are much easier to evaluate since the signal and the observations are independent. Moreover all variables $\{Y_i(x)\}_{1\leq i\leq k, x\in I}$ are independent and identically distributed Bernoulli with mean 1/2.

For any $A \in \mathcal{A}$, define $q_k(A) = E_Q(F_A(\xi_k)L_k|\mathcal{Y}_k)$. Note that, according to (7), we have, for any $B \in \mathfrak{X}$, $|B| \leq m$,

$$P\left(\cap_{x\in B}\{\xi_k(x)=1\}|\,\mathcal{Y}_k\right) = \frac{\displaystyle\sum_{A\in\mathcal{A}, A\supset B} q_k(A)}{\displaystyle\sum_{A\in\mathcal{A}} q_k(A)}.$$

Therefore the conditional law of ξ_k given $\mathcal{Y}_k$ is completely determined by the $q_k(A)$'s with $A \in \mathcal{A}$.

Our goal will be attained if one can find a recursive formula for the "unnormalized measure" q_k. To this end, set

$$D_A(y) = p_0^{|I|}\left(\frac{1-p_0}{p_0}\right)^{<y>}\left(\frac{1-p_1}{p_0}\right)^{|A\cap I|}\left(\frac{p_0p_1}{(1-p_0)(1-p_1)}\right)^{<y>_{A\cap I}},$$

where $< y >_{A\cap I} = \sum_{x\in A\cap I} y(x)$.

Using independence and identity (4), we have

$$\begin{aligned}
q_{k+1}(A) &= E_Q\left\{F_A(\xi_{k+1})L_{k+1}|\mathcal{Y}_{k+1}\right\} \\
&= E_Q\left[E_Q\left\{F_A(\xi_{k+1})\Lambda(Y_{k+1},\xi_{k+1})|Y_{k+1},\xi_k\right\}L_k|\mathcal{Y}_k\right] \\
&= D_A(Y_{k+1})E_Q\left[E_Q\left\{F_A(\xi_{k+1})|\xi_k\right\}L_k|\mathcal{Y}_k\right] \\
&= D_A(Y_{k+1})E_Q\left[\nu_{\xi_k}(A)L_k|\mathcal{Y}_k\right] \\
&= D_A(Y_{k+1})\sum_{B\in\mathcal{A}} E_Q\left[\nu_{\xi_k}(A)F_B(\xi_k)L_k|\mathcal{Y}_k\right] \\
&= D_A(Y_{k+1})\sum_{B\in\mathcal{A}} M(A,B)E_Q\left[F_B(\xi_k)L_k|\mathcal{Y}_k\right] \\
&= D_A(Y_{k+1})\sum_{B\in\mathcal{A}} M(A,B)q_k(B)
\end{aligned}$$

Therefore we obtain the so-called "Zakai" equation

$$q_{k+1}(A) = D_A(Y_{k+1})\sum_{B\in\mathcal{A}} M(A,B)q_k(B). \tag{8}$$

Note also that for any $B \in \mathfrak{X}$, $|B| \leq m$, one can define $q_k(B)$ by the formula

$$q_k(B) = \sum_{A \in \mathcal{A}, A \supset B} q_k(A).$$

It follows easily that for any such B

$$q_k(B) = E_Q(F_B(\xi_k) L_k | \mathcal{Y}_k)$$

and

$$E_P\left\{F_B(\xi_k) | \mathcal{Y}_k\right\} = P\left(\cap_{x \in B}\{\xi_k(x) = 1\} | \, \mathcal{Y}_k\right) = \frac{q_k(B)}{\sum_{A \in \mathcal{A}} q_k(A)}.$$

Remark 2.1 *Using the same technique, one can prove that for any $1 \leq k \leq n$,*

$$P(Y_k = y | \mathcal{Y}_{k-1}) = \frac{p_k(y)}{\sum_{z \in \{0,1\}^I} p_k(z)}, \tag{9}$$

where

$$p_k(y) = \sum_{A \in \mathcal{A}} \sum_{B \in \mathcal{A}} D_A(y) M(A,B) q_{k-1}(B), \quad 1 \leq k \leq n. \tag{10}$$

Note that, using (4) and (10) and the definition of $D_A(y)$, one can write

$$\begin{aligned} \sum_{z \in \{0,1\}^I} p_k(z) &= \sum_{z \in \{0,1\}^I} \sum_{A \in \mathcal{A}} \sum_{B \in \mathcal{A}} D_A(z) M(A,B) q_{k-1}(B) \\ &= \sum_{A \in \mathcal{A}} \sum_{B \in \mathcal{A}} M(A,B) q_{k-1}(B) \sum_{z \in \{0,1\}^I} D_A(z) \\ &= \sum_{A \in \mathcal{A}} \sum_{B \in \mathcal{A}} M(A,B) q_{k-1}(B). \end{aligned}$$

It follows that

$$P(Y_1 = y_1, \ldots, Y_n = y_n) = \prod_{k=1}^{n} P(Y_k = y_k | \mathcal{Y}_{k-1}) \tag{11}$$

can be written in terms of (9). This expression can be used to find maximum likelihood estimates of M and p_0, p_1.

In view of applications, one can restrict ourself to finite sets $\mathfrak{X}$. However, to be realistic, one should take $\mathfrak{X}$ bigger that I. Think about ships sailing in a certain region, where a satellite take pictures of a subregion. It is quite possible that some ships are outside of satellite reach.

Under this finiteness hypothesis, Zakai equation (8) is a finite sum and it can be evaluated easily (theoretically). Note that by definition, q_0 is determined by the initial law of the targets. Having observed Y_1, one can calculate the measure q_1, and so on.

Let us remark that some calculations can be done off-line. Probabilities $M(A, B)$, for $A, B \in \mathfrak{X}$, can be stored as a matrix, prior to starting the observation process. Note also that the computation of q_k is nothing but matrix multiplication.

The location of the m targets after the k-th observation can be estimated by choosing $A \in \mathcal{A}$, $|A| = m$, such that $q_k(A) = \max_{B\in\mathcal{A}} q_k(B)$.

3 Algorithm when the transition mechanism is unknown

Suppose now that we don't know the value of the transition function M defined in (1), and we want again to find the position of targets. Throughout the rest of the section, M will be considered as random. In that case, we have to estimate both the positions of targets and the law of M.

This case is of course a little more complicated and we need to introduce new notations. For any $A \in \mathcal{A}$ and any $k \geq 0$, define

$$\begin{aligned}
\tilde{r}_k(A) &= E(F_A(\xi_k)|\mathcal{Y}_k) = \tilde{\mu}_k(F_A) \\
r_{k+1}(A) &= E(F_A(\xi_{k+1})|\mathcal{Y}_k) = \mu_{k+1}(F_A) \\
\tilde{r}_k^{(M)}(A) &= E(F_A(\xi_k)|\mathcal{Y}_k, M) = \tilde{\mu}_k^{(M)}(F_A) \\
r_{k+1}^{(M)}(A) &= E(F_A(\xi_{k+1})|\mathcal{Y}_k, M) = \mu_{k+1}^{(M)}(F_A),
\end{aligned}$$

where $\nu(f)$ is a shorthand notation for $\displaystyle\int f(y)\nu(dy)$. Using the results of section 2, the following identities hold true:

$$\tilde{r}_k^{(M)}(A) = \frac{q_k(A)}{\sum_{B\in\mathcal{A}} q_k(B)},$$

and

$$\tilde{r}_{k+1}^{(M)}(A) = \frac{1}{C_{k+1}^{(M)}} D_A(Y_{k+1}) \sum_{B\in\mathcal{A}} M(A, B)\tilde{r}_k^{(M)}(B), \quad k \geq 0, \tag{12}$$

where

$$C_{k+1}^{(M)} = \sum_{A,B\in\mathcal{A}} D_A(Y_{k+1})M(A, B)\tilde{r}_k^{(M)}(B).$$

Note also that for any $k \geq 1$,

$$r_{k+1}^{(M)}(A) = \sum_{B \in \mathcal{A}} M(A,B)\tilde{r}_k^{(M)}(B). \tag{13}$$

Now, our aim is to find a recursive algorithm to compute $\tilde{r}_k(A)$, for any $A \in \mathcal{A}$.

Using the projection property of conditional expectations, we get

$$\begin{aligned}
\tilde{r}_{k+1}(A) &= E(E(F_A(\xi_{k+1})|\mathcal{Y}_{k+1}, M)|\mathcal{Y}_{k+1}) \\
&= E\left(\tilde{r}_{k+1}^{(M)}(A)|\mathcal{Y}_{k+1}\right) \\
&= \tilde{\eta}_{k+1}\left(\tilde{r}_{k+1}^{(\cdot)}(A)\right) \\
&= \int \tilde{r}_{k+1}^{(M)}(A)\tilde{\eta}_{k+1}(dM),
\end{aligned}$$

where $\tilde{\eta}_{k+1}$ refers to the law of M, given $\mathcal{Y}_{k+1}$.

We have now to find how to compute expectations with respect to the measure $\tilde{\eta}_{k+1}$. This is done using a useful result due to Del Moral and Miclo (2002). See [2] for a general statement of the result.

Lemma 3.1 (Del Moral-Miclo) *For any bounded measurable function ϕ of M, we have*

$$\tilde{\eta}_k(\phi) = \frac{E\left(\phi(M)\prod_{i=1}^k \mu_i^{(M)}(\Lambda(Y_i,\cdot))\,\middle|\,\mathcal{Y}_k\right)}{E\left(\prod_{i=1}^k \mu_i^{(M)}(\Lambda(Y_i,\cdot))\,\middle|\,\mathcal{Y}_k\right)}.$$

Proof

$\triangleleft$ For sake of completeness, we give a proof in our setting. First, for a given M, $(\xi_k)_{k\geq 0}$ is a Markov chain. So $(M,\xi_k)_{k\geq 0}$ is also a Markov chain. Using Kallianpur-Stridel formula, one can check that for any bounded measurable function ψ of (M,ξ_k),

$$E_P(\psi(M,\xi_k)|\mathcal{Y}_k) = \frac{E_Q(\psi(M,\xi_k)L_k|\mathcal{Y}_k)}{E_Q(L_k|\mathcal{Y}_k)},$$

where L_k is defined by (6).

Applying the formula to $\psi(M,\xi_k) = \phi(M)$, one obtains

$$\begin{aligned}
E_P(\phi(M)|\mathcal{Y}_k) &= \frac{E_Q(\phi(M)L_k|\mathcal{Y}_k)}{E_Q(L_k|\mathcal{Y}_k)} \\
&= \frac{E_Q(\phi(M)E_Q(L_k|\mathcal{Y}_k, M)|\mathcal{Y}_k)}{E_Q(E_Q(L_k|\mathcal{Y}_k, M))}.
\end{aligned}$$

It is easy to check that

$$E_Q(L_k|\mathcal{Y}_k, M) = \mu_k^{(M)}(\Lambda(Y_k, \cdot))E_Q(L_{k-1}|\mathcal{Y}_{k-1}, M),$$

so that, using induction, one gets

$$E_Q(L_k|\mathcal{Y}_k, M) = \prod_{i=1}^{k-1} \mu_i^{(M)}(\Lambda(Y_i, \cdot)),$$

yielding the result. $\triangleright$

By lemma 3.1, we have

$$\tilde{\eta}_{k+1}(\phi) = \frac{\tilde{\eta}_k\Big(\phi\mu_{k+1}^{(\cdot)}(\Lambda(Y_{k+1}, \cdot))\Big)}{\tilde{\eta}_k\Big(\mu_{k+1}^{(\cdot)}(\Lambda(Y_{k+1}, \cdot))\Big)}.$$

From the definition of $\tilde{r_k}^{(M)}$, one gets, using equality (13),

$$\begin{aligned}
\mu_{k+1}^{(M)}(\Lambda(Y_{k+1}, \cdot)) &= \sum_{A\in\mathcal{A}} D_A(Y_{k+1}) r_{k+1}^{(M)}(A) \\
&= \sum_{A,B\in\mathcal{A}} D_A(Y_{k+1}) M(A,B) \tilde{r}_k^{(M)}(B) \\
&= C_{k+1}^{(M)}.
\end{aligned}$$

Next, (12) yields

$$\begin{aligned}
\tilde{r}_{k+1}(A) &= \int \frac{1}{C_{k+1}^{(M)}} D_A(Y_{k+1}) \sum_{B\in\mathcal{A}} M(A,B)\tilde{r}_k^{(M)}(B)\ \tilde{\eta}_{k+1}(dM) \\
&= \frac{\displaystyle\int \frac{1}{C_{k+1}^{(\cdot)}} D_A(Y_{k+1}) \sum_{B\in\mathcal{A}} M(A,B)\tilde{r}_k^{(M)}(B)\mu_{k+1}^{(M)}(\Lambda(Y_{k+1}, \cdot))\ \tilde{\eta}_k(dM)}{\displaystyle\int \mu_{k+1}^{(M)}(\Lambda(Y_{k+1}, \cdot))\ \tilde{\eta}_k(dM)} \\
&= \frac{\displaystyle\int D_A(Y_{k+1}) \sum_{B\in\mathcal{A}} M(A,B)\tilde{r}_k^{(M)}(B)\ \tilde{\eta}_k(dM)}{\displaystyle\int C_{k+1}^{(M)}\ \tilde{\eta}_k(dM)}. \qquad (14)
\end{aligned}$$

Note that $\displaystyle\int C_{k+1}^{(M)}\ \tilde{\eta}_k(dM)$ is just a normalizing term.

Remark 3.1 *Applying equality (14), when the measure $\tilde{\eta}_k$ is the Dirac measure at M_0, we obtain the same formula as the one described in section 2.*

We are now in a position to describe the new algorithm.

3.1 Algorithm

Initialization: As in the first algorithm, $\tilde{r}_o$ and $\tilde{\eta}_0$ are determined by the initial law of the targets.

Recursion: For any $k \geq 0$, given $\tilde{r}_k$, $\tilde{r}_k^{(\cdot)}$, and $\tilde{\eta}_k$, compute $\tilde{r}_{k+1}^{(M)}$, using (12), $\tilde{r}_{k+1}$, using (14). Then, for any bounded measurable ϕ of M,

$$\tilde{\eta}_{k+1}(\phi) = \frac{\displaystyle\int \phi(M) \sum_{A,B\in\mathcal{A}} D_A(Y_{k+1}) M(A,B) \tilde{r}_k^{(M)}(A)\ \tilde{\eta}_k(dM)}{\displaystyle\int \sum_{A,B\in\mathcal{A}} D_A(Y_{k+1}) M(A,B) \tilde{r}_k^{(M)}(A)\ \tilde{\eta}_k(dM)}.$$

As in the first algorithm, we can locate the m targets at the k-th iteration by choosing $A \in \mathcal{A}$ such that $\tilde{r}_k(A) = \max_{B\in\mathcal{A}} \tilde{r}_k(B)$.

4 Simulations

Animations representing the results of the simulations described below can be obtained at the web site http://www.lsp.ups-tlse.fr/Fp/Gentil/ensimulations.html. Calculations were done using C++ and MATLAB.

4.1 Simulation results when M is known

When M is known, one can use the algorithm described in section 2. For the estimation of the positions of the targets, we chose the sets $A \in \mathcal{A}$ maximizing $q_k(A) = \max_{B\in\mathcal{A}} q_k(B)$. Note that several sets can satisfy this property, due to the lack of precision, especially in the first iterations.

In order to simplify calculations, we assumed that $\mathfrak{X} = I$. The Markovian targets are non-intersecting nearest neighbors random walks moving up (resp. down, right and left), with probability α_1 (respectively α_2, α_3 and $\alpha_4 = 1-\alpha_1-\alpha_2-\alpha_3$). The initial distribution q_0 was chosen to be the uniform law on all possible configurations.

For memory and computing time reasons, we restricted the simulations to the cases of one, two and three targets, and images of size 200×200 in the case of one target, size 60×60 for two targets, and size 20 for three targets.

In order to estimate the efficiency of our algorithm, we computed the mean error over several time intervals. We found out that the positions predictions were quite good after 10 to 30 steps. We also took into account various values of parameters p_0 and p_1. The error made at each iteration was calculated in the following way: in case of just one estimate A, we calculated the L^1-distance between the targets and the estimate; in case of several estimates, the largest L^1-distance was kept. The first iteration is never considered. The results are reported in Tables 1, 2 and 3.

Table 1: Mean error for one target in images of size 200×200*.*

t	$[2, 100]$	$[10, 100]$	$[30, 100]$
$p_0 = p_1 = 0.9$	8.7	4.1	1.2
$p_0 = p_1 = 0.95$	4.3	0.3	0.3

Table 2: Mean error for two targets in images of size 60×60*.*

t	$[2, 100]$	$[10, 100]$	$[30, 100]$
$p_0 = p_1 = 0.9$	6.5	1.7	1.8
$p_0 = p_1 = 0.95$	3.1	1.1	1.2

Table 3: Mean error for three targets in images of size 20×20*.*

t	$[2, 100]$	$[10, 100]$	$[30, 100]$
$p_0 = p_1 = 0.9$	4.0	3.1	2.6
$p_0 = p_1 = 0.95$	3.5	2.5	2.0

Note that in each case, the algorithm presents really satisfying results. From the 10-th, or 30-th iteration (depending on the choice of parameters p_0 and p_1), the distance between the estimation and the targets is about one or two pixels. This is due to the fact that the algorithm provides an exact solution to the resolution of the optimal filter. The estimation is of course better in the case of one target.

Remark 4.1 *A natural question to ask is how to track more than two targets for large images? One solution is to use one dimensional optimal filters to try to approximate the optimal filter for several targets. This can be done in the following way. We use the optimal filter for one target. Then we assume that the first target is on the first estimation and we use again the optimal algorithm to find another target, and so on. Tables 4, 5 and 6 below represent the results obtained by simulations.*

Table 4: Mean error for two targets with the approximation of the algorithm and in images of size 60×60*.*

t	$[2, 100]$	$[20, 100]$	$[30, 100]$
$p_0 = p_1 = 0.9$	*7.9*	*3.9*	*4.4*
$p_0 = p_1 = 0.95$	*4.1*	*0.9*	*0.6*

Table 5: Mean error for two targets with the approximation of the algorithm in images of size 200×200*.*

t	$[2, 100]$	$[10, 100]$	$[30, 100]$
$p_0 = p_1 = 0.9$	*47*	*42*	*45.3*
$p_0 = p_1 = 0.95$	*5.7*	*0.5*	*0.6*
$p_0 = p_1 = 0.99$	*20.1*	*17.3*	*19.0*

Table 6: Mean error for three targets with the approximation of the algorithm in images of size 200×200*.*

t	$[2, 100]$	$[10, 100]$	$[30, 100]$
$p_0 = p_1 = 0.9$	*40.7*	*35.6*	*31*
$p_0 = p_1 = 0.95$	*45.9*	*45.3*	*44.1*
$p_0 = p_1 = 0.99$	*15.1*	*14.1*	*14.7*

These results are quite interesting. For two targets and images of size 60×60*, the mean errors are almost the same as the one obtained in Table 2. However, as can be seen in Tables 5 and 6, the stability of the approximation of the optimal filter is not as good. Remark that for three targets, when* $p_0 = p_1 = 0.99$*, the approximation is quite satisfactory.*

4.2 Simulation results when M is unknown

We now present simulations using the algorithm developed in section 3. For sake of simplicity, we chose to simulate only one target in images of size 200×200.

As in the previous subsection, we use the uniform law on all pixels. The measure $\tilde{\eta}_0$ is a discrete measure with N Dirac measures defined as

$$\tilde{\eta}_0(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \sum_{i=0}^{N} \frac{1}{N+1} \delta_{\left(\frac{i}{2N}, \frac{i}{2N}, \frac{1}{2} - \frac{i}{2N}, \frac{1}{2} - \frac{i}{2N}\right)}(\alpha_1, \alpha_2, \alpha_3, \alpha_4). \tag{15}$$

This is not the best choice for $\tilde{\eta}_0$ but it is very simple for computations and the results obtained are quite interesting. Of course, in simulations, the motion of the target is random and is not in the support of $\tilde{\eta}_0$ defined by (15).

For the simulations, we used several values for N and (p_0, p_1). The results are reported in the Table 7.

Table 7: *Mean error for one target in images of size* 200×200 *for random* M*.*

$p_0 = p_1 = 0.9$	$[2, 100]$	$[10, 100]$	$[30, 100]$
$N = 2$	36	27	20
$N = 10$	21	10.5	0.9
$N = 100$	20	8	0.8
$p_0 = p_1 = 0.95$	$[2, 100]$	$[10, 100]$	$[30, 100]$
$N = 2$	6.5	1.4	0.5
$N = 10$	7.1	0.1	0.1
$N = 100$	7.1	0.1	0.1

From these results, one can see that when the measure is initialized by a discrete measure with $N = 10$ or 100, the approximations look like simulations when we know the transition function M of the targets. When $N = 2$, the measure $\tilde{\eta}_0$ is too bad to estimate the target.

Remark 4.2 *In view of applications, one can either consider that the parameters* p_0 *and* p_1 *have been estimated or one can use the maximum likelihood method, (e.g. using 11) to estimate them from the observation of images.*

5 Conclusion

We can conclude that the optimal filter is quite easy to implement whether the transition function M of the targets is known or unknown. Simulations results reported in the previous section showed quite convincing arguments in favour of the proposed approach.

On the negative side, the optimal filter method need a lot of memory to compute predictions for 3 targets or 2 targets in large images.

A good solution could be to use an hybrid method combining the optimal filter for one or two targets and "Interacting Particle methods". See e.g. Del Moral and Miclo [3] for an interesting review of the subject. With this hybrid method used in the section 3, we could compute also the optimal filter to find the positions and the number of targets. This is still work in progress.

Finally, one could replace the hypothesis of conditional independence of the observations by introducing Gibbs measure noise (e.g. Rémillard and Beaudoin 1999 for other applications of Gibbs noise [4]). The only difference would be in the definition of the L_k's.

Acknowledgments

Partial funding in support of this work was provided by the Natural Sciences and Engineering Research Council of Canada, by the MITACS-PINTS Network of Center of Excellence, Lockheed Martin Canada and by the Institut de finance mathématique de Montréal.

References

[1] Ballantyne, D. J., Chan, H. Y. and Kouriztin, M. A. (2002). A branching particle-based nonlinear filter for multi-target tracking. Submitted.

[2] Del Moral, P. (2002). *Feynman-Kac formulae genealogical and interacting particle models*, 24th Finish Summer School on Probability Theory, Rolf Nevanlinna Institute, Lahti.

[3] Del Moral, P. and Miclo, L. (2000). *Branching and Interacting Particle Systems Approximations of Feynman-Kac Formulae with Applications to Non-Linear Filtering*, Séminaire de Probabilités XXXIV, Lectures Notes in Mathematics, Springer-Verlag Berlin, Vol 1729, 1–145.

[4] Rémillard, B. and Beaudoin, C. Statistical comparison of images using Gibbs random fields. In *Vision Interface'99*, 612–617 (1999).