

# Capteurs Infra-Rouges pour la Reconstruction 2D

## Modélisation Markovienne et filtrage binaire

A. Chebira, R. Reynaud, T. Maurin

IEF, Bat 220 Université Paris Sud F-91405 Orsay Cedex. FRANCE

### RÉSUMÉ

Le problème abordé ici est la définition d'un débruitage sur des signaux binaires en utilisant une modélisation markovienne de ces signaux. Pour chaque pixel binaire, la matrice de transition dépend du voisinage de ce pixel, la chaîne de Markov engendrée est donc inhomogène. L'originalité est dans la définition de ce voisinage qui exploite le voisinage spatial et temporel pour prendre en compte a priori sur l'inertie et la rigidité des objets observés et aussi dans la détermination des matrices de transition.

### 1. INTRODUCTION

Nous disposons de signaux binaires et bruités et le problème est d'obtenir un traitement de complexité abordable par des moyens de calcul embarqués sur un véhicule pour éliminer du bruit sur ces signaux. Cette phase du traitement doit être relativement robuste, car elle est suivie d'une étape de triangulation de complexité variable en fonction des signaux d'entrée et donc évidemment très sensible à la qualité des signaux obtenus par l'étape de débruitage.

Bien que nous disposions finalement d'un modèle de la scène à reconstruire, dont nous nous servons après coup, nous voulons dans l'étape de débruitage introduire une information a priori qui ne soit pas trop contraignante comme dans le cas d'un modèle paramétrique, et nous avons choisi d'utiliser une modélisation markovienne de ces signaux sans modèle d'état associé.

Pour chaque pixel binaire, la chaîne de Markov engendrée est inhomogène et est contrôlé localement par le choix de la matrice de transition parmi une série de transitions possibles. La complexité est évidemment plus faible mais l'efficacité du débruitage provient de la définition de ce contexte local et de la détermination des matrices de transition. Nous présentons au §2 l'application puis nous montrons comment nous passons de notre a priori physique à la détermination du contexte local au §3 et des matrices de transition au §4, enfin des résultats situent les performances de cette étape de débruitage de l'algorithme.

### 2. APPLICATION

Dans la littérature scientifique, nombreuses sont les tentatives de résolution du difficile problème de la vision artificielle. La plupart de ces systèmes sont construits autour de caméras CCD. Le but recherché est la construction d'un modèle géométrique de la scène qui soit adapté à un traitement par ordinateur pour faire de la

### ABSTRACT

We define a filtering method to remove noise on binary signals by using a Markov modelization of these signals. For each binary pixel, transition matrix depends from its neighbourhood, and the generated Markov chain is then inhomogeneous. The original idea is in the neighbourhood definition that uses spatial and temporal neighbours to take into account the notion of inertia or rigidity of the observed obstacles, and also in the determination of the transition matrices.

reconnaissance de formes, de la détection et de l'évitement d'obstacle, du guidage d'engins autonomes, etc...

On veut construire un modèle 2D d'une scène observée par des capteurs infrarouges spécifiques [1]. On se situe dans le domaine automobile, un véhicule dit "intelligent" est équipé de capteurs infrarouges, ainsi que de capteurs d'autres types, de réseaux locaux et d'ordinateurs standards. L'objectif fixé est la détection d'obstacles dans le but d'assistance à la conduite automobile. Les capteurs utilisés renvoient des images lignes binaires. Par rapport aux images que délivrerait une caméra CCD dans la même situation, les données renvoyées par les capteurs véhiculent une quantité d'information beaucoup plus faible.

Notre challenge [2] se situe justement au niveau de l'abondance d'information acquise par les capteurs. On a choisi des capteurs linéaires dont on espère extraire le même niveau sémantique d'information que dans le cas d'une image CCD. L'abondance d'information amenant une redondance, il n'est pas possible de faire mieux que dans le cas d'une image CCD, mais notre but est de montrer qu'il est possible de faire aussi bien, plus rapide, moins complexe et moins cher. Pour cela nous modifions le compromis initial d'un traitement intensif et bas niveau sur une information redondante et abondante vers un compromis différent entre un traitement de plus haut niveau sémantique avec ajout d'a priori et une information moins abondante mais, dont on espère, qu'elle est toujours suffisamment pertinente pour l'application que nous visons.

L'ensemble du traitement, c'est-à-dire débruitage, triangulation, pistage et reconstruction de la scène, est contrôlé à chaque étape du traitement par un jeu d'hyperparamètres [4]. Dans le cas de l'algorithme de débruitage, c'est le jeu des matrices de transition qui jouent le rôle des hyperparamètres qui servent à contrôler le traitement de débruitage. C'est pourquoi nous insistons sur la signification physique et la détermination de ces matrices.



### 3. ALGORITHME DE DÉBRUITAGE

On se propose ici d'ajouter un deuxième niveau de décision pour corriger les erreurs de décision du capteur. Pour cela on utilise le fait que l'échantillonnage temporel se fait avec un pas de 0.1 seconde. Une inertie est alors imposée à l'évolution temporelle des images délivrées par le capteur.

L'a priori que l'on a sur les objets observés par les capteurs et leurs dynamiques nous permet de déduire les deux contraintes suivantes:

1- les objets observés par les capteurs ayant une dynamique finie et connue, une interprétation logique doit être observée pour les décisions prises à des *intervalles de temps consécutifs*.

2- les objets observés par les capteurs sont supposés rigides, de dimensions constantes, un couplage entre les décisions sur l'état *de deux bits voisins existe*.

#### 3.1 Inertie temporelle

Pour exploiter la première contrainte on suppose que l'état d'un bit à l'instant  $t$  ne dépend que de son état à l'instant précédent ( $t-1$ ). En d'autres termes on suppose que le processus qui génère le bit est Markovien [5]. A chaque position angulaire  $\alpha$  et pour chaque instant  $t$  est associée une variable aléatoire binaire  $I(\alpha, t)$  qui peut prendre deux états "1" et "0". A chaque état est affectée une probabilité  $p(\alpha, t) = P(I(\alpha, t)=1)$ . Un vecteur à deux dimensions est affecté à chaque position angulaire  $\alpha$ . Il contient [3] les probabilités que le bit correspondant à l'angle  $\alpha$  et à l'instant  $t$  soit égale à 1 ou à 0. Si l'on suppose que l'on connaisse la probabilité initiale, de l'instant zéro et la matrice de transition, on peut prédire l'état que va prendre un bit pour tout instant  $t > 1$ . Cette représentation est classique, on l'utilise pour effectuer une transformation d'image ligne binaire en image ligne à "niveaux de gris", la probabilité associée à chaque pixel est vue comme un niveau de gris. On augmente en effet l'entropie de l'image 1D.

$$\text{Angle } \alpha \rightarrow P(\alpha, t) = [ p(\alpha, t), 1 - p(\alpha, t) ] \quad (1)$$

L'hypothèse que le processus est Markovien nous permet d'écrire que la probabilité de l'état à l'instant  $t+1$  est égale à la probabilité de l'état à l'instant  $t$  multipliée par une matrice de transition  $T$ . C'est à dire:

$$P(\alpha, t+1) = \begin{bmatrix} p(\alpha, t+1) \\ 1 - p(\alpha, t+1) \end{bmatrix}^t = P(\alpha, t) \cdot \begin{bmatrix} P(0/0) & P(1/0) \\ P(0/1) & P(1/1) \end{bmatrix} \quad (2)$$

$P(U/V)$  est la probabilité de transition de l'état  $U$  à l'état  $V$ ,  $U$  et  $V$  prennent les valeurs 0 ou 1. Les conditions suivantes doivent être vérifiées:

$$P(0/0) + P(1/0) = 1 \quad (3)$$

$$P(0/1) + P(1/1) = 1 \quad (4)$$

#### 3.2 Inertie spatio-temporelle

Chaque capteur fournit une image ligne binaire à chaque tour que l'on notera image 1-D. On dispose de deux a priori sur le problème: -l'inertie traduisant une dynamique finie des objets évoluant dans la scène -et la rigidité des obstacles. L'évolution de l'image 1-D doit satisfaire aux contraintes induites par ces deux a priori. Les objets observés par les capteurs peuvent:

- s'éloigner

- se rapprocher
- subir une translation latérale
- apparaître ou disparaître.

Dans le cas où un objet s'éloigne ou se rapproche, l'angle solide sous lequel il est observé diminue ou augmente, le nombre de bits qui passent à un dans une région angulaire augmente ou diminue. Dans le cas d'une translation latérale, l'image d'un obstacle dans l'espace 1-D subit une translation. Dans le cas où un obstacle rentre ou sort du champ de vision du capteur, des bits passent à un ou à zéro.

Ce sont des notions analogues aux notions de pistage et poursuite de cibles. L'idée est de constituer à partir des observations 1-D une représentation temps-angle, de définir une zone d'intérêt dans cette image associée au traitement d'un pixel (en grisé sur la figure 1). La profondeur de cette zone représente la mémoire que l'on conserve, la pente de la frontière de la zone traduit la dynamique que l'on autorise sur les obstacles. Mais les traitements deviennent alors trop complexes pour les moyens de calcul disponibles, et nous avons décidé de conserver la mémoire du système au travers au travers de la grandeur analogique  $p(\alpha, t)$  (figure 3). L'évolution de cette grandeur est alors commandée par un contexte local à chaque pixel au travers de la transition appliquée pour obtenir  $p(\alpha, t+1)$ .

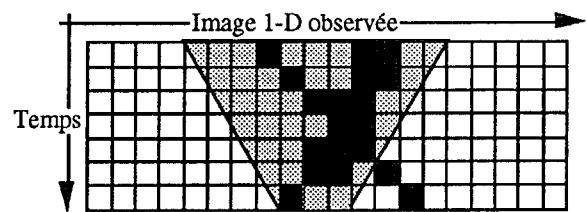


Figure 1 - Représentation temps-angle

L'ensemble de ces 6 pixels que nous allons définir est appelé le contexte local associé au pixel  $i$  et comporte une information sur le voisinage spatial du pixel en question et une information sur une trace temporelle concernant l'information passée relative à ce pixel. Pour chaque pixel  $i$  d'une image observée à l'instant  $t+1$ , le contexte comprend (figure 2) : le pixel ; les deux pixels voisins immédiats dans l'image 1-D (pour conserver une isotropie des traitements [6]); les trois pixels correspondant dans l'image filtrée à l'instant  $t$  (et la grandeur analogique  $p(\alpha, t)$  associée pour maintenir une mémoire d'une certaine profondeur).

Le choix d'une matrice de transition se fait sur un critère de contexte local. On peut alors faire 64 interprétations sémantiques possibles ( $2^6$  contextes différents possibles). Vue les symétries et les cas particuliers, le nombre de contextes peut être ramené à 12. A chaque contexte est rattachée une matrice de transition pré-calculée qui traduit une contrainte imposée à l'évolution de l'image 1-D au travers de l'évolution de la probabilité  $P(\alpha, t+1)$ .

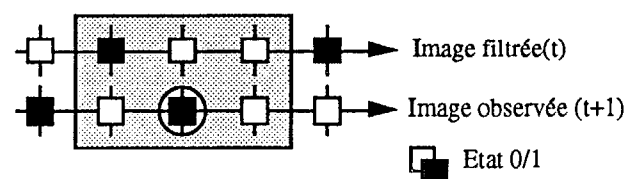


Figure 2 - Contexte local d'un pixel de l'image

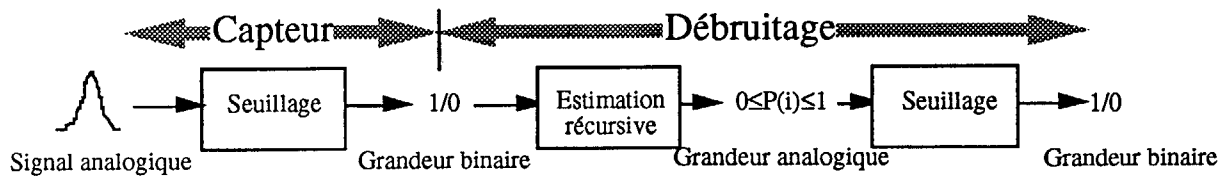


Figure 3 : Algorithme de filtrage des signaux binaires

### 3.3 Algorithme de filtrage

On analyse le contexte local d'un pixel  $i$  (Cadre grisé de la figure 2) qui détermine alors une matrice de transition pré-définie. Cette transition est appliquée au vecteur de probabilité  $P(i,t)$  du pixel  $i$  à l'instant  $t$  pour inférer la probabilité de son état à l'instant  $t+1$ . Par seuillage de la probabilité, on déduit l'état du bit  $i$  (1 ou 0) que l'on va utiliser récursivement pour l'itération suivante.

```

/* On dispose de  $P_i(i=1)$ , de l'image filtrée à l'instant  $t$ , et de
l'image observée par le capteur à l'instant  $t+1$ . */
Pour chaque position angulaire  $i$ 
  *Evaluation de la configuration du contexte local
  *Sélection d'une matrice de transition correspondante
  *Application de la matrice pour calculer la nouvelle
  probabilité
  *Définition par seuillage de l'état du bit  $i$ 
Fin Pour.
  
```

### 3.4 Exemple d'évolution

Il est à signaler que l'état inféré peut être différent de l'état renvoyé par le capteur. Par exemple l'état inféré peut rester égal à 1 pour un pixel observé égal à 0 si la transition ne fait pas décroître suffisamment la probabilité pour la faire tomber au-dessous du seuil (qui est fixé à 0,5 sur le tableau suivant) et ceci dépend évidemment de ses 5 voisins et des matrices de transition associées à toutes les configurations.

	$\alpha-1$	$\alpha$	$\alpha+1$	$P(\alpha,t)$
Filtrée(t) Observée(t+1)	1	1	1	0,9
	↓			
Filtrée(t+1) Observée(t+2)	1	1	1	0,71
	↓			
Filtrée(t+2) Observée(t+3)	1	1	1	0,52
	↓			
Filtrée(t+3)	1	0	1	0,33

Tableau 1 - Exemple d'évolution du pixel  $\alpha$

## 4. DÉTERMINATION DES MATRICES DE TRANSITION

La difficulté de l'algorithme proposé est l'estimation des différentes matrices de transitions. Plusieurs méthodes peuvent être utilisées pour cela. On peut introduire les a priori directement en développant l'équation (2) ou utiliser des images lignes sans bruit et avec bruit simulées comme séquences d'apprentissage. La disponibilité de séquences d'apprentissage permet en effet d'utiliser des méthodes d'apprentissage supervisé.

### 4.1 Apprentissage sur exemples

Une formulation du problème sous forme d'équation d'état permettra en utilisant le critère de vraisemblance d'estimer les paramètres de la chaîne de Markov. Le problème est simplifié du moment où l'on dispose de séquences d'apprentissage [7]. Une autre voie est l'utilisation de réseaux de neurones comme filtre binaire, les poids synaptiques déterminés après la phase d'apprentissage peuvent alors être comparés aux valeurs des matrices de transitions, une interprétation des poids synaptiques en termes de valeurs de transition peut alors être envisagée. Ces deux voies vont être explorées et comparées dans le futur.

### 4.2 Détermination en fonction de l'inertie

Pour chaque contexte, nous savons s'il y a ou non une transition potentielle du pixel  $i$ . Les matrices de transition vont servir à régler la vitesse de convergence vers un état stationnaire et l'efficacité de la méthode provient du rapport des différentes vitesses de convergence qui dépendent du contexte. C'est pourquoi nous revenons sur la détermination des puissances successives  $N$  de la matrice  $T$  qui va nous donner une relation entre  $a$ ,  $b$  (5) et  $N$  réglant une inertie sur la transition du pixel.

$$T = \begin{bmatrix} a & 1-a \\ b & 1-b \end{bmatrix} \quad (5)$$

on a:

$$T^N = \begin{bmatrix} \frac{(a-b)^N - a(a-b)^N + b}{1-a+b} & \frac{(1-a)(1-(a-b)^N)}{1-a+b} \\ \frac{(1-(a-b)^N) * b}{1-a+b} & \frac{1-a+(a-b)^N * b}{1-a+b} \end{bmatrix} \quad (6)$$

1- Cas  $a=b$ : La matrice  $T$  est singulière et on a les deux relations suivantes

$$T^N = T \quad (7)$$

$$P(\alpha, t+1) = P(\alpha, t) \cdot T = \begin{bmatrix} a \\ 1-a \end{bmatrix}^t \quad (8)$$

La relation (8) nous permet de dire que l'état à un instant  $t$  quelconque ne dépend que de la matrice de transition. A partir de n'importe quel état initial  $P_0$ , on converge dès la première application de la matrice de transition vers une mesure stationnaire définie à l'avance. La propriété de processus Markovien est perdue. Le choix d'une telle matrice de transition peut être intéressante si l'on est sûr de la décision que l'on veut prendre. On peut forcer l'état d'un bit à un ou à zéro en choisissant respectivement  $T_1$  ou  $T_0$  données par:

$$T_1 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad T_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad (9)$$

2- Cas  $a \neq b$ : La matrice de transition  $T$  est régulière, on a la propriété suivante:



$$\lim_{N \rightarrow \infty} T^N = \begin{pmatrix} b & 1-a \\ 1-a+b & 1-a+b \\ b & 1-a \\ 1-a+b & 1-a+b \end{pmatrix} = T_{\infty} \quad (10)$$

La matrice  $T_{\infty}$  est singulière, et possède les mêmes propriétés que dans le cas  $a=b$ . L'application d'une matrice de transition une infinité de fois à un état initial quelconque, le fait converger vers un état stationnaire qui est défini par le choix des paramètres  $a$  et  $b$ . La vitesse de convergence est définie par l'écart entre  $a$  et  $b$ . Cette propriété va nous permettre de contrôler l'inertie pour le passage d'un état quelconque à un état stationnaire défini à l'avance.

#### 4.3 Exemple de calcul d'une matrice de transition

Considérons le calcul de la matrice de transition pour le passage de l'état "1" à l'état "0" dans le cas du contexte initial du tableau 1. La probabilité de l'état initial est  $p(\alpha,0)$  et on suppose que sa valeur est 0.9, ce qui traduit une incertitude résiduelle sur la connaissance de cet état. La probabilité de l'état final est  $p(\alpha,N)$  où  $N$  est le nombre de coups nécessaire pour le passage de l'état initial à un état final possédant la même incertitude résiduelle. On prend par exemple  $N=6$  et  $p(\alpha,N)=0.1$ . On écrit alors  $P(\alpha,N) = P(\alpha,0) T^N$ , on résout cette équation en utilisant (6) pour obtenir les valeurs de  $a$  et  $b$ . Ce qui donne:

$$a = 0,85 \text{ et } b = 0,55 \quad (11)$$

### 5. PRÉSENTATION DES RÉSULTATS

La figure 4 représente l'image binaire 1-D d'un obstacle qui subit une translation latérale obtenue par simulation sans bruit. L'axe vertical représente le temps, l'axe transversal représente les positions angulaires. La figure 5 représente cette même image bruitée obtenue par transition aléatoire avec une probabilité égale à 0.85, ce qui correspond à un rapport signal/bruit élevé. La figure 6 représente l'image filtrée, qui est utilisée par les étages ultérieurs de traitement. On remarque la nette amélioration apportée par l'algorithme. Cet exemple a été choisi pour montrer l'efficacité de l'algorithme qui reste très simple à implanter et d'une complexité très faible.

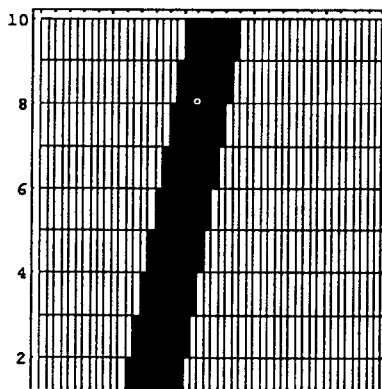


Figure 4 - Signal Sans bruit

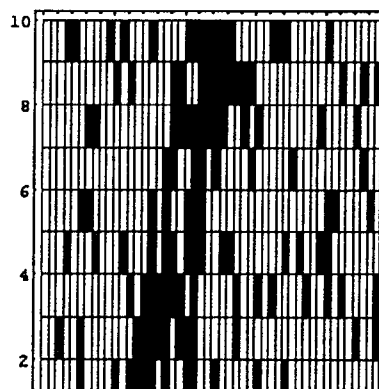


Figure 5 - Signal Bruité

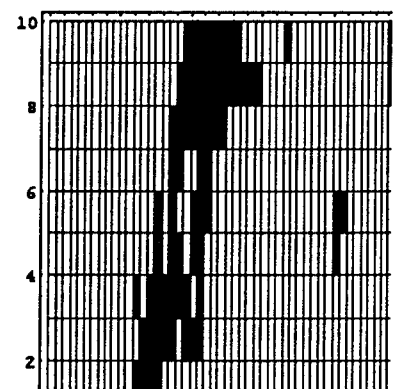


Figure 6 - Signal débruité

### 6. CONCLUSION

On a présenté un algorithme de filtrage de signaux binaires, cet algorithme procède en deux étapes, transformation de l'image binaire en image analogique en utilisant une modélisation Markovienne, l'image obtenue est alors seuillée pour obtenir l'image débruitée. Le formalisme adopté pourrait permettre aussi de prendre en compte le cas où une partie de l'image n'est pas disponible (données manquantes ou données en retard[]), ou le cas où on conserve une indétermination sur l'image filtrée. Cela passe par la définition d'un troisième état ("inconnu" pour l'image observée et "indéterminé" pour l'image filtrée) en plus des deux états "0" et "1", ce qui augmente les dimensions des matrices de transitions, accroît la complexité du problème mais ne modifie pas la formulation choisie.

### 7. RÉFÉRENCES

- [1] A. Chebira, R. Reynaud, T. Maurin, " Reconstruction of 2D Parameters using Infrared Sensors" SPIE's 1993 International Symposium on Optics, Imaging, and Instrumentation, San Diego, California USA, 11-16 july 1993.
- [2] A. Chebira, R. Reynaud et G. Demoment "Fusion de données multicapteurs: application à la détection d'obstacles en temps réel", Colloque GRETSI, pp 73-76, Juan les Pins, Septembre 1991.
- [3] P. Bremond "Modélisation Markovienne", Editions ENSTA, 1987 I.S.B.N. 2.7225.0763.3.
- [4] A. Chebira, R. Reynaud, T. Maurin "Description d'une méthode d'estimation 2D, utilisation conjointe de techniques floues et probabilistes" Rapport interne I.E.F, université de Paris XI, à paraître en octobre 1993.
- [5] P. Bouhemy et F. Heitz "Champs markoviens : fondement et application à l'analyse de séquence d'images" Journées thématiques Approches markoviennes en signal et images, pp20-43, 22-23 septembre 1992.
- [6] W. QIAN, D. M. Titterington "Multidimensional Markov Chain Models for Image Textures" J. R. Statist. Soc. B, 53, N°3, 661-674.
- [7] L. R. Rabiner et B. H. Juang "An introduction to hidden Markov models", IEEE ASSP Magazine January 86, 4-16.