

Une architecture alternative pour la mise en oeuvre temps réel d'un algorithme markovien de détection de mouvement

Christophe DUMONTIER, Franck LUTHON et Jean-Pierre CHARRAS

Laboratoire de Traitement d'Images et Reconnaissance de Formes

Institut National Polytechnique de Grenoble

46, Avenue Félix Viallet, 38031 Grenoble Cedex

e-mail : luthon@tirf.inpg.fr

RÉSUMÉ

Dans cet article, nous présentons dans un premier temps un algorithme simple et robuste de détection de mouvement, basé sur une approche markovienne, et dans un deuxième temps une architecture matérielle spécifique pour son implantation en temps réel. Les mises en oeuvre de ce type d'algorithmes sont généralement réalisées sur des machines parallèles. De récents travaux proposent aussi l'implantation d'algorithmes simplifiés sur réseaux résistifs intégrés sur des circuits VLSI. Après l'étude détaillée de notre algorithme, nous proposons une solution alternative de mise en oeuvre, validée par le développement d'une carte prototype intégrant l'ensemble du processus de détection (de l'acquisition vidéo à la visualisation des masques finaux). Une cadence de traitement de l'ordre de 15 images par seconde est obtenue.

ABSTRACT

In this paper, a simple and robust MRF-based motion detection algorithm is presented and a hardware architecture is proposed for its real time implementation. Implementations of this kind of algorithms are generally made on parallel machines. Recent researches also propose analog VLSI implementations of simplified MRF-based algorithms. According to the detailed analysis of our algorithm, we propose another alternative, based on the development of an image processing board integrating the whole process (from video acquisition to mask visualization). A processing rate of 15 images per second on 128 x 128 pixel images can be achieved by our board prototype.

1 Introduction

Notre équipe a développé un algorithme robuste de détection de mouvement, basé sur une approche markovienne [2]. La qualité des résultats obtenus nous a amenés à étudier la mise en oeuvre temps réel de cet algorithme en vue d'applications industrielles (télésurveillance, suivi d'objets).

Dans un premier temps, une analyse détaillée de l'algorithme a été effectuée afin de décomposer celui-ci en **blocs fonctionnels** et d'évaluer la **complexité calculatoire** de chacun de ces blocs. Par la suite, une **architecture adaptée** a été déduite de cette analyse, et une première validation sur une carte générique de traitement d'images a été réalisée [2]. Le développement d'une carte plus spécifique a ensuite été mené à bien, aboutissant à un traitement à une cadence de 15 images par seconde.

2 Définition de l'algorithme de détection

Le but de l'algorithme est d'affecter à chaque point de l'image une étiquette, *fixe* ou *mobile*. Nous faisons l'hypothèse que le champ d'étiquettes binaires *cachées* à un instant donné t (noté E_t) est un champ de Markov défini relativement à un voisinage ν d'ordre 2 (cf. figure 1). La solution finale correspond à la configuration la plus probable du champ de primitives (éti-

quettes) étant donné le champ des données (observations).

Les observations ici retenues sont représentées par la valeur absolue de la différence temporelle de la fonction de luminosité entre deux images successives : $O_t = |I_t - I_{t-1}|$.

Le critère d'estimation utilisé pour trouver la configuration finale est le Maximum A Posteriori (MAP) :

$$Pr[E/O] \text{ maxi} \implies Pr[E]Pr[O/E] \text{ maxi} \quad (1)$$

En pratique, maximiser cette probabilité revient à minimiser une fonction d'énergie globale U , constituée de deux termes : un terme d'énergie d'attache aux données U_a et un terme d'énergie associée au modèle U_m .

Le lien entre observations et étiquettes est défini par :

$$o_s = \Psi(e_s) + n \quad (2)$$

où

$$\Psi(e_s) = \begin{cases} 0 & \text{si } e_s = b \\ \alpha > 0 & \text{si } e_s = a \end{cases}$$

n représente un bruit gaussien de variance σ^2 (variance des observations) et α , la moyenne des observations non nulles.

Le terme énergétique U_a est directement issu de la relation (2) :

$$U_a(o, s) = \frac{1}{2\sigma^2} \sum_{s \in S} [o_s - \Psi(e_s)]^2 \quad (3)$$

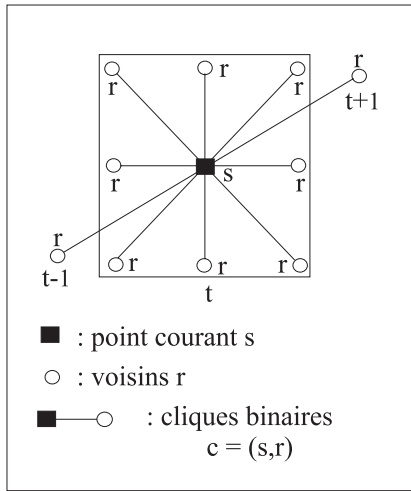


FIG. 1 — Voisinage et cliques binaires associées de notre modèle.

Le terme d'énergie U_m traduit la probabilité a priori du champ markovien des étiquettes E :

$$U_m(e) = \sum_{c \in C} V_c(e_s, e_r)$$

C représente l'ensemble des cliques c définies par le voisinage spatio-temporel de la figure 1. Cette probabilité fait appel à la définition de potentiels élémentaires à niveau $V_c(e_s, e_r)$ dépendant uniquement des étiquettes de la clique considérée :

$$V_c(e_s, e_r) = \begin{cases} -\beta & \text{si } e_s = e_r \\ +\beta & \text{si } e_s \neq e_r \end{cases}$$

La valeur du paramètre β dépend de la clique considérée. Trois paramètres sont ici utilisés, β_p associé à la clique passée, β_s associé aux cliques spatiales et β_f associé à la clique future. Enfin, la recherche de la configuration du champ d'étiquettes E_t minimisant la fonction d'énergie globale U est réalisée en utilisant l'algorithme de relaxation déterministe sous-optimal des ICM. La technique de visite de sites retenue lors de cette phase de relaxation est le parcours **séquentiel** (i.e. ligne par ligne) de chacune des images.

A noter que l'utilisation du futur pour l'obtention des masques à l'instant t engendre un retard d'une image dans l'apparition des résultats. La figure 2 fournit trois résultats de détection illustrant les trois principales propriétés de notre algorithme.

3 Analyse détaillée de l'algorithme

A l'heure actuelle, les implantations temps réel d'algorithmes markoviens sont réalisées sur deux types de systèmes : les machines parallèles qui exploitent le caractère intrinsèquement parallèle de la modélisation markovienne [1, 5] et les réseaux de neurones qui représentent une solution attractive pour la minimisation d'une fonction d'énergie [6]. Toutefois, ces solutions ne débouchent que très rarement sur des réalisations opérationnelles pour des raisons d'encombrement, de complexité et de coût de tels systèmes. Il nous est apparu intéressant d'analyser, pour un algorithme donné, la complexité

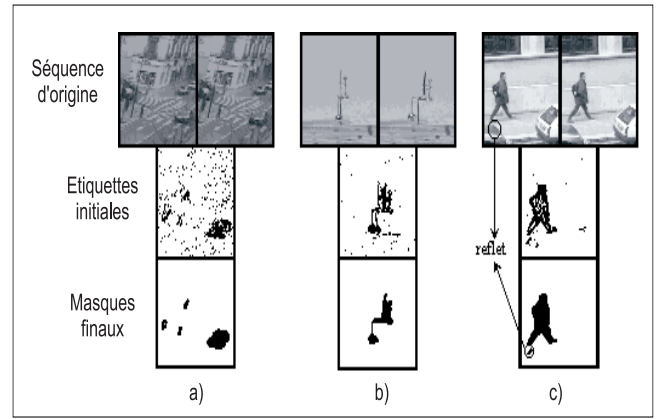


FIG. 2 — Propriétés de l'algorithme : a) élimination du bruit. b) élimination de l'écho. c) précision des masques.

calculatoire réelle mise en jeu et, à partir de cette analyse, de rechercher la meilleure adéquation entre l'algorithme et l'architecture de la machine cible.

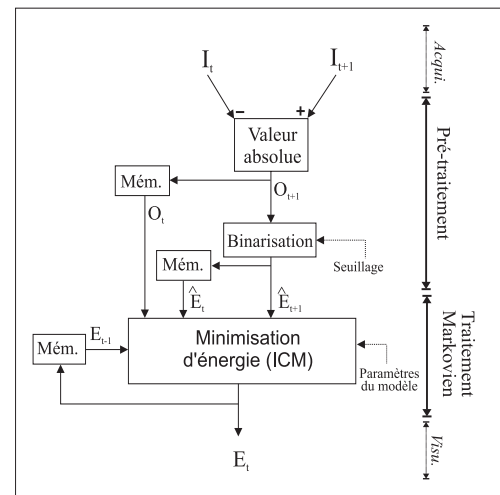


FIG. 3 — Synoptique de l'algorithme de détection et séparation en blocs fonctionnels.

Le synoptique de notre algorithme de détection est donné par la figure 3. Deux blocs fonctionnels principaux se dégagent de ce synoptique :

- un bloc de **pré-traitement** calculant le champ d'observations O_t et le champ initial des étiquettes \hat{E}_t à partir de deux images successives de la séquence traitée ;
- un bloc de **minimisation d'énergie** permettant d'obtenir le champ final des étiquettes E_t , c'est-à-dire les masques des objets mobiles.

Dans le cadre d'une application concrète, il faut ajouter à ces deux blocs, un bloc visant à fournir les données initiales au bloc de pré-traitement (i.e. acquisition vidéo) et un bloc pour visualiser les résultats (i.e. visualisation vidéo).

La complexité calculatoire de ces différents blocs n'est évidemment pas la même. L'acquisition et la visualisation ne sont

que de simples opérations de conversion. Le pré-traitement effectuée des opérations élémentaires de différence et de seuillage d'images. Par contre, le bloc de minimisation d'énergie est plus complexe :

- les calculs à effectuer font appel à des opérations flottantes,
- la quantité de calculs exécutés en chaque point est considérable (utilisation d'un voisinage d'ordre 2),
- le caractère itératif de l'algorithme des ICM nécessite d'effectuer plusieurs balayages de chaque image avant d'atteindre la convergence.

Une analyse précise de cette phase de traitement nous a permis d'évaluer les différents temps de calcul affectés à l'évaluation des deux termes énergétiques :

1. calcul de U_a : 25 %,
2. calcul de U_m : 75 %.

L'étude des calculs exécutés lors de l'évaluation de ces deux termes a permis de constater la présence de redondances ainsi que des analogies entre la nature de ces calculs et des opérations simples de convolution associées à la lecture d'une table de correspondance (c.f. figure 4). La prise en compte de ces remarques a abouti à une diminution d'un facteur 2 du temps de calcul global [4].

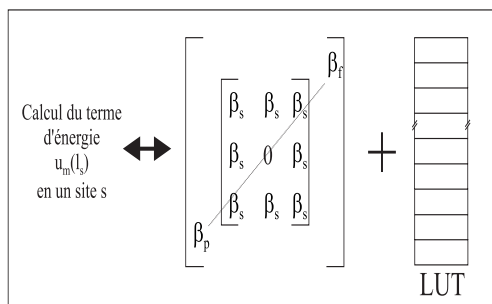


FIG. 4 — Analogie "convolution + LUT" lors du calcul du terme u_m en un site s .

4 Architecture proposée

La structure générale du système qui se dessine après cette première analyse est une architecture de type *pipeline*. Les opérations de conversion, de différence et de seuillage peuvent être réalisés en temps réel (i.e. à la cadence de 25 ima/s) sur le flux vidéo sans aucun problème par l'intermédiaire de solutions *câblées*.

De par la complexité des calculs mis en jeu lors de la phase de relaxation markovienne, une solution câblée n'est pas envisageable. Un traitement *asynchrone* a donc été retenu, permettant de réaliser l'ensemble de ces opérations sans ralentir la cadence générale de traitement du système. Ce bloc nécessite l'introduction d'une zone d'échange de données ayant deux rôles essentiels :

- d'une part, permettre au bloc de minimisation de venir chercher l'ensemble des données utiles à la détermination des masques et de stocker les résultats de ces calculs,
- d'autre part, d'assurer la continuité du flux vidéo entre l'entrée et la sortie de notre système.

A partir de ces constatations, une mise en oeuvre de notre algorithme sur une architecture alternative a été proposée, architecture associant des modules fonctionnant en mode *pipeline* et un module fonctionnant en mode *asynchrone*.

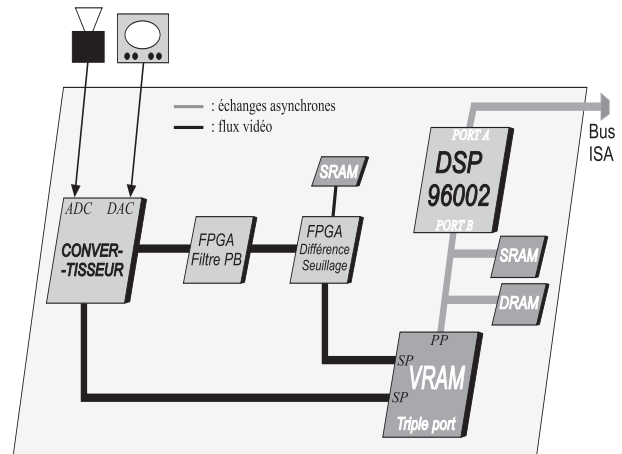


FIG. 5 — Architecture générale de la carte.

5 Réalisation pratique

La figure 5 donne le synoptique général de la carte développée. On retrouve l'architecture mixte *pipeline/asynchrone* énoncée précédemment. Les blocs d'acquisition, de visualisation et de pré-traitement travaillent sur le flux vidéo. Ces opérations sont effectuées par des composants standards (convertisseurs, composants logiques programmables).

Le bloc de relaxation markovienne a été implanté sur un processeur de signal (DSP). Ce choix s'explique par la nature des opérations à effectuer (convolutions + LUT). De plus, l'architecture interne de ce type de processeur permet d'introduire une dose de parallélisme dans l'exécution des calculs. La zone d'échange (mémoire tampon) est physiquement représentée par une mémoire vidéo triple ports.

La photo de la figure 6 montre la carte prototype développée, formant un système de détection de mouvement :

- **complet**, la gestion des E/S vidéo étant intégrée sur la carte,
- **autonome**, aucune donnée n'étant échangée avec l'extérieur durant la phase de traitement,
- **compact et peu coûteux**, la carte au format PC-ISA n'utilisant que des composants standards du marché.

La cadence de calcul atteinte est de l'ordre de **15 ima/s** sur des images de taille 128 x 128 points. Une modification de la technique de visite des sites lors de la relaxation permet d'atteindre une cadence de 20 ima/s sans dégradation notable

des masques obtenus [4]. Ces résultats sont à rapprocher avec ceux obtenus par [3] lors de l'implantation du même algorithme sur une machine parallèle SIMD comportant 256 processeurs, et qui a permis d'atteindre une cadence de traitement de l'ordre de 10 ima/s.

Le traitement d'images 256 x 256 points est possible en sous-échantillonnant les images initiales : un filtre passe-bas a été prévu à cet effet et s'intercale entre le bloc d'acquisition et le bloc de pré-traitement. Les masques finaux sont alors reconstruits par simple interpolation (recopie de la valeur du pixel courant). Cette interpolation engendre toutefois une légère perte dans la précision des masques.

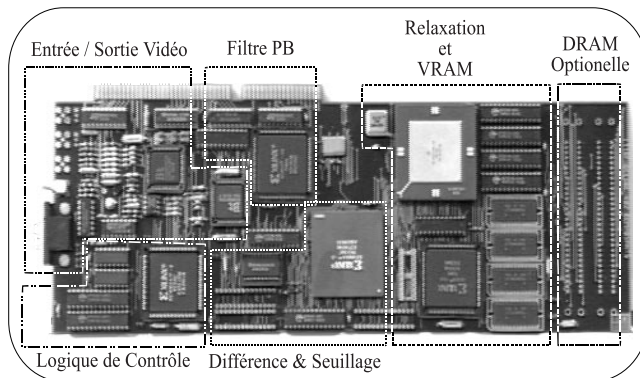


FIG. 6 — Photo du prototype réalisé.

6 Conclusion

Ces travaux montrent que la mise en oeuvre temps réel d'un algorithme markovien peut être réalisée autrement que sur des machines parallèles, et que le développement de circuits spécifiques (fort coûteux) n'est pas indispensable. Des systèmes relativement simples permettent de nos jours de réaliser de telles implantations, à condition d'analyser précisément l'algorithme considéré et de rechercher la meilleure solution architecturale pour son implantation.

A l'heure actuelle, l'extension de la détection binaire à une détection *multi-étiquettes* est à l'étude. Le but est alors d'affecter une étiquette différente à chacun des objets mobiles présent dans la scène observée. La méthode envisagée consiste à ajouter une phase finale de multi-étiquetage après la phase de détection binaire. La mise en oeuvre de cette extension ne nécessite aucune modification matérielle du prototype développé et engendrera une diminution d'un facteur 2 de la cadence de calcul.

Références

- [1] A. Bellon, J.P. Derutin, F. Heitz, and Y. Ricquebourg. Real-time collision avoidance at road crossings on board the Prometheus-ProLab 2 vehicle. *Intelligent Vehicle*, Octobre 1994.
- [2] A. Caplier, C. Dumontier, F. Luthon, and P.Y. Coulon. Algorithme de détection de mouvement par modélisation

markovienne. Mise en oeuvre sur DSP. *Traitement du Signal*, Vol.13(2) :177–190, 1996.

- [3] A. Caplier, F. Luthon, and C. Dumontier. Real time implementations of an MRF-based motion detection algorithm. *Accepted to be published in Journal of Real Time Imaging, special issue on Real Time Motion Analysis*, 1997.
- [4] C. Dumontier. *Etude et mise en oeuvre temps réel d'un algorithme markovien de détection de mouvement par approche markovienne*. PhD thesis, Institut National Polytechnique, Grenoble, France, Novembre 1996.
- [5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6) :721–741, Novembre 1984.
- [6] C. Koch, J. Marroquin, and A. Yuille. Analog neuronal networks in early vision. *Proc. Natl. Acad. Sci., USA, Biophysics*, Vol.83, :4263–4267, Juin 1986.