

ALGORITHME MARKOVIEEN DE DETECTION DE MOUVEMENT. MISES EN OEUVRE TEMPS REEL ?

Alice CAPLIER, Franck LUTHON, Christophe DUMONTIER
Laboratoire de Traitement d'Images et Reconnaissance de Formes
Institut National Polytechnique de Grenoble
LTIRF, INPG, 46 avenue Félix-Viallet
38031 GRENOBLE Cedex, FRANCE
Email : name@tirf.inpg.fr

Résumé

Le handicap de beaucoup d'algorithmes de traitement d'images est la lourdeur des calculs. Les algorithmes basés sur la modélisation markovienne n'échappent pas à cet inconvénient, bien au contraire. Ayant mis au point un algorithme markovien dédié à la détection de mouvement, nous examinons les solutions matérielles possibles pour atteindre une cadence de traitement compatible avec les exigences temps réel. Trois implantations (machine parallèle, carte de traitement d'images à base de DSP et réseau résistif VLSI analogique) sont décrites puis comparées.

Abstract

The main problem of most image processing techniques is the computational cost. MRF-based processing particularly requires a significant charge of computations. This paper studies and compares three solutions (SIMD machine, DSP-based image processing board, and VLSI analog resistive network) to implement an MRF-based motion detection algorithm.

1 Introduction

L'efficacité de la modélisation markovienne pour régulariser des problèmes mal posés du traitement d'images a été largement prouvée. La définition d'algorithmes mettant en jeu des modèles markoviens conduit inmanquablement à la minimisation d'une fonction d'énergie par des algorithmes de relaxation coûteux en temps de calcul. Or une donnée essentielle en analyse de scènes dynamiques demeure le traitement temps réel. Cette notion ne fait pas nécessairement référence à la cadence vidéo de 25 images/seconde mais plutôt au temps de traitement maximal admissible vis-à-vis d'une application donnée.

Après avoir rappelé les grandes lignes d'un algorithme markovien de détection de mouvement [1], nous étudions les trois solutions matérielles envisagées afin d'accélérer la cadence des traitements. Ayant constaté que la programmation de cet algorithme sur une station de travail classique ne nous permet pas d'espérer un traitement suffisamment rapide, nous nous sommes tournés vers trois solutions : la programmation d'une machine parallèle, la mise en oeuvre sur une carte de traitement d'images à base de DSP et la réalisation d'un réseau résistif VLSI analogique. Les caractéristiques de chaque réalisation sont exposées,

l'objectif est de répertorier les avantages et les inconvénients de chaque solution.

2 Modèle de détection

La détection est envisagée comme un problème d'étiquetage statistique binaire : il s'agit de déterminer si un pixel est fixe (étiquette "0") ou mobile (étiquette "1"). Le champ global E des étiquettes à estimer sur une image est supposé être markovien relativement au voisinage et aux cliques de la figure 1.

Dans le cas où la caméra est fixe et l'illumination de la scène est constante, les variations temporelles de la fonction de luminance sont dues au mouvement d'un objet. Les informations de mouvement ou champ O d'observations issues de la séquence sont définies en chaque site $s = (x, y)$ par : $o_t(s) = |I_t(s) - I_{t-1}(s)|$.

La détection de mouvement vise à rechercher le champ E le plus probable étant donné le champ O ce qui est équivalent à la minimisation d'une fonction d'énergie globale U effectuée par l'algorithme itératif de relaxation déterministe des ICM. Le synoptique de la figure 2 résume le fonctionnement général de l'algorithme. Programmé en langage C sur une station de travail Sun SPARC-10, cet algorithme de-

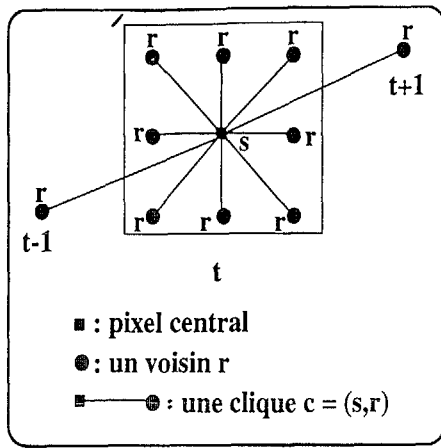


Figure 1: Voisinage et cliques associées

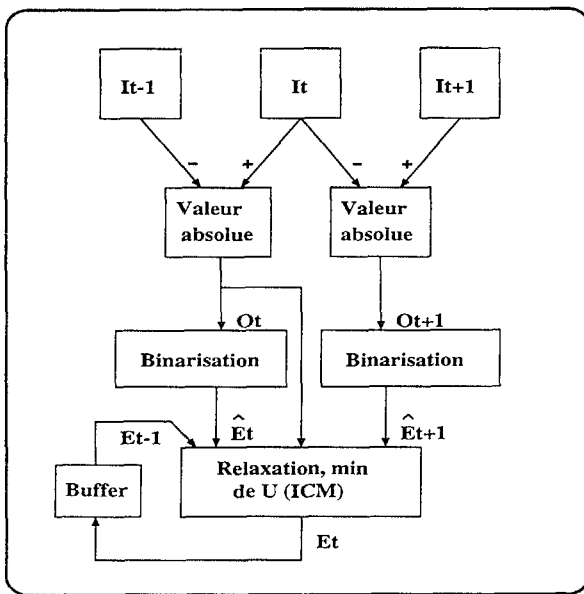


Figure 2: Algorithme de détection de mouvement

mande en moyenne 5s pour analyser une image de taille 128x128 (soit 380ms pour une itération et 5 itérations en moyenne avant la convergence). Ce temps n'est absolument pas compatible avec les exigences du traitement temps réel.

3 Mise en oeuvre sur une machine parallèle : la CNAPS

Une caractéristique essentielle de la modélisation markovienne réside dans la localité des calculs qu'elle engendre : le choix de l'étiquette du pixel s ne dépend que des étiquettes de ses voisins. Les calculs étant locaux et identiques pour tous les sites, ils sont fortement parallélisables. D'où l'idée naturelle d'exploiter cette propriété en implantant notre algorithme sur une machine multi-processeur.

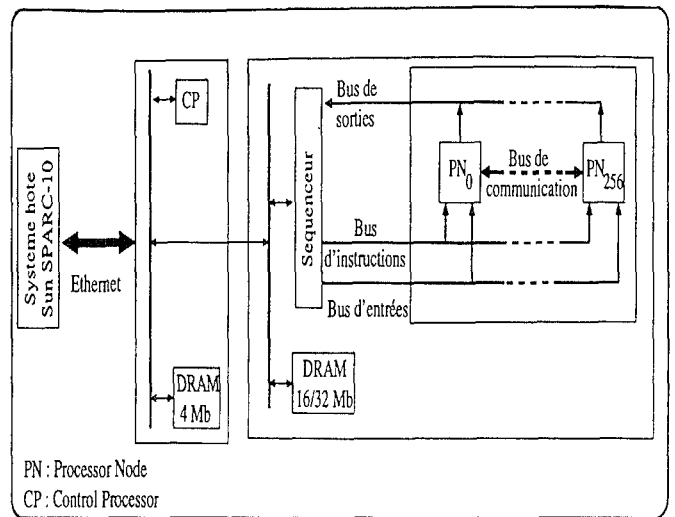


Figure 3: Architecture de la CNAPS

3.1 La machine CNAPS

La CNAPS, développée par la société Advance Haute Technologie, est une machine parallèle SIMD (*Single Instruction Multiple Data*) constituée d'un réseau linéaire de 256 processeurs (cf. figure 3). Elle communique avec une station hôte Sun SPARC-10 via le réseau Ethernet. Chaque processeur dispose de 4Ko de mémoire locale et réalise des opérations sur des nombres entiers ou à virgule fixe. La programmation de la machine peut se faire soit directement en assembleur (ce qui est très performant mais aussi très fastidieux) soit en langage C parallèle (ce qui est moins efficace mais beaucoup plus aisé pour les habitués du langage C ANSI).

3.2 Mise en oeuvre de l'algorithme

L'architecture de la CNAPS oblige avant tout à penser au déroulement parallèle de l'algorithme de détection de mouvement. Du fait des calculs intrinsèquement parallèles, cette tâche est aisée. Le problème le plus délicat est la répartition des données sur chaque processeur de la machine sachant que la taille de la mémoire locale est limitée. Nous avons adopté une répartition uniforme des données en attribuant une ligne d'images par processeur. Afin de limiter les transferts interprocesseurs coûteux en temps d'exécution, la prise en compte d'une modification éventuelle d'étiquette a lieu image par image et non point par point comme dans la version séquentielle. La conséquence est la propagation moins rapide des contraintes markoviennes et donc l'augmentation relative du nombre moyen d'itérations avant la convergence (15 en moyenne au lieu de 5). Comme sur beaucoup de

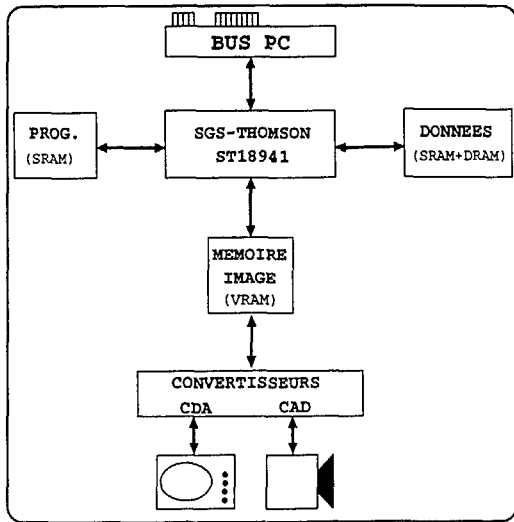


Figure 4: Synoptique de la carte VPC941

machines parallèles, la gestion des entrées/sorties demeure séquentielle.

Pour une image de taille 128x128, la cadence de traitement atteinte est de **10 images/seconde** (soit 6.5ms par itération et 15 itérations en moyenne).

4 Carte générique de traitement d'images à base de DSP

D'emblée, l'intérêt d'utiliser un unique DSP pour accélérer les temps de traitement de notre algorithme n'est pas évident. En effet, avec un seul DSP, il n'est pas question d'exploiter le parallélisme des calculs. Néanmoins, un examen plus approfondi de la quantité de calcul, du type d'opérations (additions et tests comparatifs essentiellement) et du type de variables mises en jeu (entiers codés sur 8 bits pour la plupart) nous a permis de valider ce choix. L'algorithme a été programmé sur la carte VPC941 de la société SECAD-SA équipée d'un DSP ST18941 de SGS-Thomson cadencé à 10MHz [4]. Le synoptique de la carte est donné sur la figure 4.

La mise en oeuvre logicielle a été faite en assembleur car aucun compilateur C fiable et optimal n'est disponible. La phase de prétraitement relative au calcul des observations fonctionne en temps réel. Ce n'est pas le cas de la partie relaxation markovienne proprement dite. La cadence finale atteinte pour la chaîne complète (depuis l'acquisition des images jusqu'aux masques des objets mobiles) est de **3 images/seconde** pour une taille de 128x128. Nous avons évalué pouvoir obtenir une cadence de **12 images/seconde** avec un DSP cadencé à 50MHz. Ce qui ressort de cette étude, c'est que le développement d'une carte à base de DSP dédiée à notre algorithme

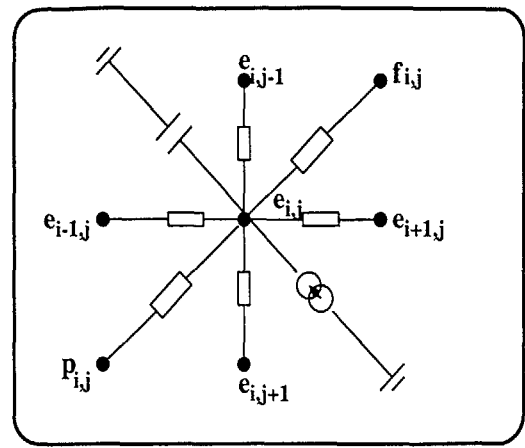


Figure 5: Cellule de Markov élémentaire

de détection de mouvement permettrait de s'approcher de la cadence vidéo.

5 Réseau résistif

Une autre solution, plus compatible avec la notion de calculs associés à un voisinage, est l'approche basée sur les réseaux de neurones. Il a été montré d'une part, que les réseaux linéaires représentent un moyen naturel de résoudre le problème de la minimisation d'une fonction d'énergie et d'autre part qu'à tout problème de minimisation d'une fonction d'énergie quadratique, il est possible d'associer un réseau électrique correspondant.

Notre premier travail a été de transformer le modèle de détection de mouvement discret à étiquettes binaires ("0" ou "1") en un modèle à étiquettes continues dans $[0, 1]$ pour lequel la fonction d'énergie globale U est quadratique tout en conservant les propriétés d'homogénéité spatio-temporelles des solutions [5]. En effet, les étiquettes du champ recherché correspondent désormais à des potentiels électriques ($V_{cc} = "0" = \text{masse} = 0V$ et $V_{cc} = "1" = 5V$). Une caractéristique essentielle du modèle obtenu est qu'il fonctionne avec deux images consécutives au lieu de trois. En effet, la fonction d'énergie U étant convexe, il n'est pas indispensable d'initialiser avec soin la recherche du minimum. Ceci a permis de réduire considérablement la taille du réseau final.

Nous avons ensuite déduit la cellule électrique élémentaire à associer à chaque pixel (cf. figure 5) et l'architecture globale du réseau (cf. figure 6). Les simulations ont témoigné du bon fonctionnement du réseau [6], il reste maintenant à se lancer dans la réalisation proprement dite. Point essentiel, **la relaxation du réseau est extrêmement rapide** (de l'ordre de quelques dizaines de nanosecondes).

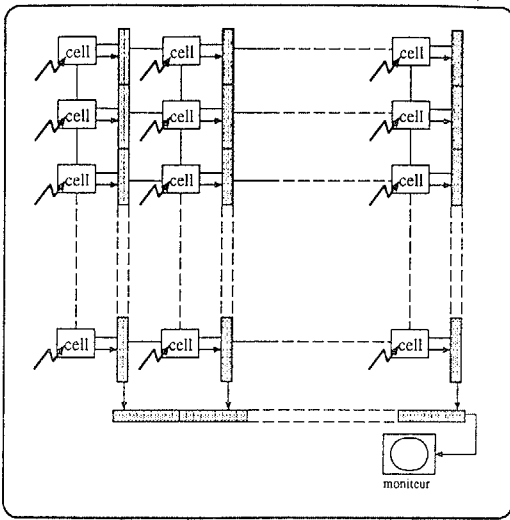


Figure 6: Architecture du réseau

6 Bilan comparatif

La contrainte temps réel nous a incité à envisager trois réalisations matérielles différentes pour notre algorithme markovien de détection de mouvement. Quelle est a posteriori la solution la plus satisfaisante ?

La solution machine parallèle permet d'exploiter naturellement le parallélisme des calculs; elle conduit à une cadence de traitement honorable et offre le double avantage de la flexibilité et de la facilité de mise en oeuvre. En revanche, la gestion des entrées/sorties reste séquentielle et dans le cadre d'applications embarquées, elle est beaucoup trop encombrante. Par ailleurs, son prix est loin d'être négligeable.

Solution a priori peu canonique, la carte de traitement d'images à base de DSP s'est révélée être une réalisation tout à fait intéressante tant au niveau de la cadence de traitement (pour peu que l'on prenne un DSP un peu plus performant) que de l'encombrement ou du prix de revient. En revanche, la mise en oeuvre n'est pas du tout flexible (en particulier, il faut tout modifier dès que la taille des images analysées varie) et la programmation de la carte en assembleur est un peu fastidieuse.

La solution réseau résistif analogique VLSI est la plus prometteuse en ce qui concerne la cadence de traitement et l'encombrement. De plus, elle intègre parfaitement les notions de calculs parallèles et de voisinage. En revanche, la gestion des entrées/sorties sur le réseau est un problème délicat; de même, la puissance dissipée est importante.

7 Conclusion

L'utilisation de la modélisation markovienne en traitement d'images doit être accompagnée de l'implantation des algorithmes sur des réalisations matérielles particulières. Pour notre part, nous avons étudié trois mises en oeuvre différentes pour un algorithme markovien de détection de mouvement. Les trois solutions peuvent être ici retenues, les cadences de traitement étant suffisantes pour l'analyse de scènes de trafic routier. En revanche, pour un algorithme markovien plus complexe, nous pensons que c'est la solution réseau résistif qui doit être envisagée car pour la machine parallèle, la taille mémoire limitée de chaque processeur pose rapidement des problèmes inextricables, quant à la carte à base de DSP, elle n'exploite pas le parallélisme des calculs.

References

- [1] F. LUTHON, A. CAPLIER "Motion Detection and Segmentation in image sequences using Markov Random Fields Modelling". In 4th Eurographics Animation and Simulation Workshop, Barcelona, Spain, September 1993, pp. 265-275.
- [2] J. BESAG "On the Statistical Analysis of Dirty Pictures". In Journal Royal Statistical Society, Vol. B-48, N.3, 1986, pp. 259-302.
- [3] S. GEMAN, D. GEMAN "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In IEEE Trans. Pattern Anal. and Machine Intel. Vol. PAMI-6, N.6, November 1984, pp. 721-741.
- [4] A. CAPLIER, C. DUMONTIER, F. LUTHON, P.Y. COULON "Algorithme de segmentation de mouvement par modélisation markovienne. Mise en oeuvre sur DSP." 20 pages, submitted to Traitement du Signal, December 1994.
- [5] F. LUTHON, V.G. POPESCU, A. CAPLIER "An MRF based motion detection algorithm implemented on analog resistive network". In ECCV'94 Proc., Stockholm, Sweden, May 1994, pp. 167-174.
- [6] F. LUTHON, S. POPESCU "Un circuit intégré analogique pour la détection de mouvement". In Proc. Journées d'études et d'échanges "Nouvelles techniques pour la compression et la représentation des signaux audiovisuels", CCETT, Rennes, France, 12-13 January 1995, pp. 105-112.