

Détection de mouvement adaptative bas niveau

A. VERDANT¹, A. DUPRET², H. MATHIAS², P. VILLARD¹

¹CEA LETI - MINATEC, 17 rue des Martyrs, 38054 Grenoble Cedex 9

arnaud.verdant@cea.fr, patrick.villard@cea.fr

²IEF, Bâtiment 220, Université de Paris 11, 91405 Orsay Cedex

antoine.dupret@ief.u-psud.fr, herve.mathias@ief.u-psud.fr

Résumé – Un algorithme de détection de mouvement bas niveau, destiné à être implanté au sein d'un capteur d'images CMOS, est présenté. Un tel algorithme doit à la fois être robuste et compatible avec une grande variété d'environnements, tout en ne requérant pas d'importantes ressources de calcul ou de mémoire dans le cas d'une implantation pour des imageurs CMOS embarqués. Destiné à un fonctionnement sans intervention humaine, cet algorithme réalise une segmentation d'objets en mouvement avec seuillage adaptatif. Une étude comparative avec un algorithme de référence est également présentée.

Abstract – A low-level motion detection algorithm for CMOS image sensor is presented. Aimed algorithm designed to be implemented in such embedded system has to be both robust and compliant to various environments while being computationally efficient. Algorithms must meet these constraints under some resources limitations, since in CMOS image sensors for example, few computational resources are available. This algorithm performs image segmentation on static video flows with adaptive thresholding and no constant to determine *a priori*.

1. Introduction

L'identification d'objets en mouvement dans un environnement fortement perturbé peut parfois sembler être une tâche complexe, tant les éléments non pertinents de la scène sont importants (e.g. bruissement de feuillage). Nos travaux portent sur un système dédié à la vidéo surveillance embarquée autonome. Il s'agit typiquement de faire apparaître sur les images acquises les objets en mouvement. Dans ce contexte, la consommation de puissance, les performances du capteur d'images et la qualité de la détection des objets sont des éléments clés. Le système développé repose sur une architecture de traitement parallèle SIMD proche de celle présentée dans [1] et associée à une matrice de pixels de type CMOS APS (Active Pixel Sensor) standard. Dans ce papier, nous présentons un algorithme qui s'implémente sur l'architecture précitée et qui répond aux contraintes de performances et d'embarquabilité requises par l'application.

De nombreux travaux ont été menés sur la vidéosurveillance, par exemple [2], où un ensemble d'approches conventionnelles pour la segmentation du mouvement sont décrites. L'actualisation du fond de la scène observée est notamment une tâche essentielle car les algorithmes temps réels intégrés aux systèmes embarqués doivent être performants dans un large éventail de situations et doivent alors adapter leur sensibilité à la

scène. La segmentation d'objets avec différence au fond a été étudiée dans [3] et [4]. Le seuillage adaptatif pour la détection de mouvement en extérieur a également été étudiée dans [5].

Ces différentes approches sont performantes mais requièrent un nombre important d'opérations à effectuer. Un algorithme robuste n'impliquant pas de ressources calcul ou mémoire importantes est alors nécessaire. De plus, un seuil local adaptatif est préférable dans le but de filtrer les éléments perturbateurs de la scène. Dans [6], un ensemble d'algorithmes de détection de mouvement bas niveau sont présentés et dans [7], un algorithme robuste et compact pour rétines artificielles, basé sur la modulation $\Sigma-\Delta$, est décrit. Ces algorithmes nécessitent un ajustement de certaines variables en fonction de l'environnement observé. Nous avons développé un algorithme original sans réglage réalisant la segmentation d'objets en mouvement alliant robustesse et « compatibilité niveau transistor » au sein d'un imageur CMOS.

Nous présenterons en partie 2 ce nouvel algorithme. La partie 3 expose une étude comparative entre cet algorithme et un algorithme de référence [7]. Une technique de seuillage adaptatif est aussi présentée dans cette partie.

2. Architecture

L'architecture programmable envisagée en tant que support des algorithmes est une machine SIMD basée sur un traitement analogique des informations [1] (figure 1).

Cette architecture est composée d'une matrice de $A \times B$ photorécepteurs à laquelle est associée une matrice de $A \times (mB)$ points mémoires analogiques, m représentant le nombre de points mémoire par pixel. La matrice ainsi formée est bordée d'un côté par un vecteur de A processeurs analogiques à capacités commutées. Une colonne de multiplexeurs permet la sélection des colonnes de macropixels ou mémoires à traiter par ces processeurs. Un séquenceur digital (CPU) délivre les instructions successives des processeurs.

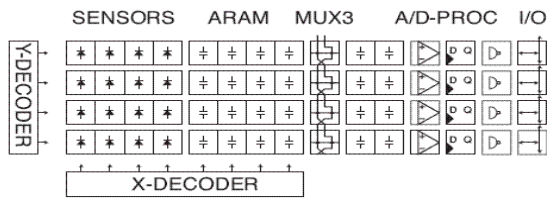


FIG. 1 : Architecture

Les processeurs analogiques, dont le principe de fonctionnement est détaillé dans [1], fonctionnent à temps discrets, et permettent des opérations telles que :

$$Acc \leftarrow a \times Acc + b \times Reg, \text{ if } Acc \leftarrow |Acc|, \dots$$

L'efficacité d'une telle architecture d'un point de vue consommation a pu être mise en avant dans [8] dès lors qu'on envisage l'implantation d'algorithmes faiblement itératifs, à noyau spatial ou temporel réduits, en utilisant une structure de calculateur équivalente à un convertisseur Analogique/Numérique.

3. Algorithme adaptatif

3.1 Contraintes de consommation

Le problème de la détection de mouvement consiste à distinguer au sein des séquences d'images les informations pertinentes du "bruit" ambiant.

Dans notre schéma de surveillance, nous souhaitons réaliser une détection de mouvement avec des capteurs d'images CMOS autonomes abandonnées, dans des environnements inconnus. Dans une telle configuration; les algorithmes doivent satisfaire à de fortes contraintes de robustesse et d'adaptabilité. Les algorithmes Markoviens sont généralement utilisés pour faire face à ces situations. Cependant, en prenant en compte la consommation de puissance et les ressources de calcul et de mémoire disponibles, ces algorithmes doivent être simplifiés tout en préservant la robustesse de la détection.

Nous avons pris pour référence l'algorithme présenté dans [7], réalisant une détection de mouvement à partir d'opérations non linéaires non intensives d'un point de vue itérations. Cet algorithme Σ - Δ suit le modèle markovien utilisé pour des implantations temps réel [9]. Cependant, la

prédétermination d'une constante N est requise, réduisant l'adaptabilité de l'algorithme.

3.2 Algorithme

3.2.1 Description de l'algorithme

Nous nous penchons ici sur un algorithme capable de réaliser une segmentation du mouvement sur des séquences pleine résolution sans détermination de constantes a priori. Tenant compte des contraintes de ressources disponibles au sein d'un imageur CMOS, les approches markoviennes ont été écartées.

Dans un premier temps, afin de détecter les variations de niveaux de gris résultant du mouvement, deux types de filtres peuvent être utilisés dans notre algorithme. Le premier est un nouveau filtre extrayant la différence absolue Δn_1 entre la valeur courante du signal S_n et celle du dernier extremum (figure 2). Le second filtre est basé sur l'estimation du fond présenté dans [7], réalisée à partir d'opérations non linéaires. La valeur estimée du fond dans [7] est utilisée pour le calcul de la différence absolue avec S_n pour chaque pixel (Δn_2).

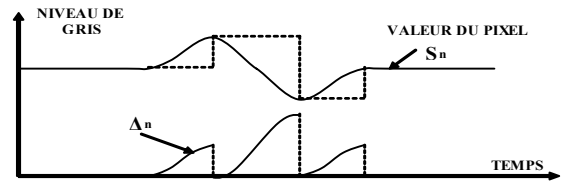


FIG. 2 : Extraction des variations temporelles (Δn_1)

L'indicateur de mouvement utilisé Δ_n ($=\Delta n_1$ ou Δn_2) est ensuite seuillé de manière adaptative suivant la technique exposée dans les équations (1) à (4) et illustré figure 3. Tout d'abord, $M1_n$ (« estimée » des variations de Δ_n) est calculée (1), pouvant être assimilée à l'écart moyen des variations du signal. $M2_n$ (« estimée » de $M1_n \times \Delta_n$) est ensuite déterminée (2). Une troisième variable $M3_n$ est finalement calculée (3) avec comme pas d'incrément la valeur $M2_n$ précédemment calculée. Cette variable oscille autour de la valeur du signal. On estime qu'il y a mouvement lorsque la différence absolue entre $M3_n$ et S_n devient supérieure à $M2_n$ (4), i.e. la variation courante du signal est supérieure à l'estimation $M2_n$. Hormis le niveau d'incrément de $M1_n$ et $M2_n$ fixé à un niveau de gris, cet algorithme réalisé à partir d'opérations non linéaires ne possède aucune constante.

$$\text{si } M1_{n-1} < \Delta_n \rightarrow (M1_n = M1_{n-1} + 1) \quad (1)$$

$$\text{sin on si } M1_{n-1} > \Delta_n \rightarrow (M1_n = M1_{n-1} - 1)$$

$$\text{si } M2_{n-1} < M1_n \times \Delta_n \rightarrow (M2_n = M2_{n-1} + 1) \quad (2)$$

$$\text{sin on si } M2_{n-1} > M1_n \times \Delta_n \rightarrow (M2_n = M2_{n-1} - 1)$$

$$\text{si } M3_{n-1} < S_n \rightarrow (M3_n = M3_{n-1} + M2_n) \quad (3)$$

$$\text{sin on si } M3_{n-1} > S_n \rightarrow (M3_n = M3_{n-1} - M2_n)$$

$$\text{si } |M3_n - S_n| > M2_n \rightarrow \text{mouvement} \quad (4)$$

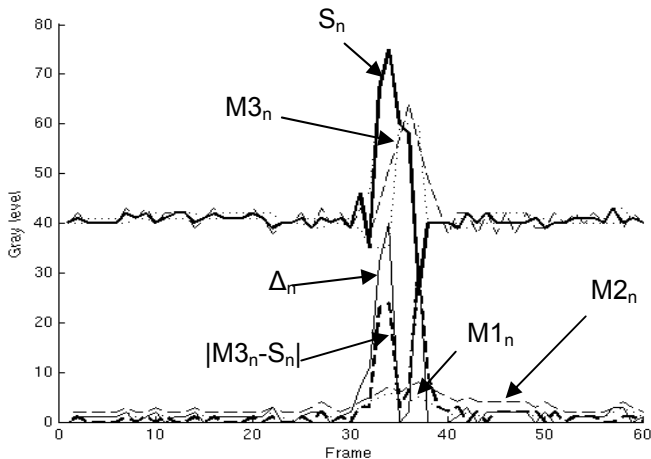


FIG. 3 : Traitement du signal temporel d'un pixel

Dans le but de réduire les faux positifs induits par les zones bruitées de l'image, un filtre supplémentaire a été ajouté. Dérivé d'une approche markovienne déterministe, ce filtre ne considère comme mouvement pertinent que les variations de niveau de gris de pixels constituant une translation locale. Ainsi, une variation du signal temporel d'un pixel considérée comme un mouvement (4) sera effectivement considérée comme un mouvement pertinent si elle succède les variations antérieures des pixels voisins dans une direction données (N, NW, W, SW, S, SE, E, NE) (figure 4).

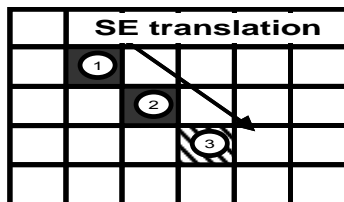
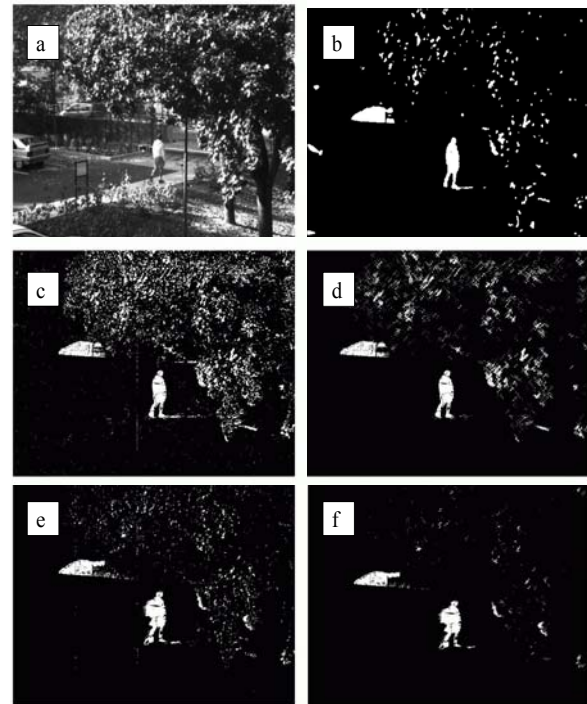


FIG. 4 : Détection de translation locales. Le mouvement est successivement détecté dans les pixels 1 et 2. Seul le pixel 3 pourra être considéré comme un mouvement pertinent dans les trames suivantes.

3.2.2 Illustrations

La figure 5 illustre la segmentation du mouvement obtenue avec l'algorithme présenté : 'a' est l'image originale, 'b' est la segmentation obtenue à partir de l'algorithme Σ - Δ combiné à Markov et des filtres morphologiques [10] ; 'c' et 'd' sont les segmentations réalisées avec l'algorithme Σ - Δ [7] (N=5), respectivement avec et sans filtre de détection de translation locale. Les images 'e' et 'f' résultant des simulations effectuées avec l'algorithme présenté, respectivement avec et sans filtre de détection de translation locale. Avec notre algorithme, le bruit du feuillage est clairement atténué.

FIG. 5 : Segmentation de la séquence *Walk*

4. Résultats

Les simulations de détection de mouvement avec l'algorithme présenté ont été réalisées sous MATLAB®. Différentes séquences présentant une grande variété de conditions intérieures et extérieures ont été testées : *Walk* (bruissement de feuillage) (séquence IEF, figure 5), *Hall Monitor*, *Pets 2002* (lumière oscillante), *dtneu_schnee* (chute de neige) et *kwbB* (i21www.ira.uka.de) (respectivement a, b, c et d sur la figure 6).

Nous avons utilisé les mesures de performance basées sur [11] et présentées tableau 1 pour l'algorithme Σ - Δ de référence et tableaux 2 et 3 pour notre algorithme : taux de détection ((DR=TP/(TP+FN)), taux de fausses alarmes (FAR=FP/(TP+FP)), et taux de faux positifs (FPR=FP/(FP+TN)) avec TP, TN, FP, FN les vrais ou faux positifs ou négatifs.

Le tableau 1 présente les résultats issus des simulations effectuées avec l'algorithme Σ - Δ (avec le paramètre N fixé à 5, cette valeur typique ayant été choisie indépendamment des caractéristiques de la scène comme ce serait le cas dans le cas d'un capteur abandonné). Les tableaux 2 et 3 illustrent les mesures effectuées avec notre algorithme.

Malgré un taux de détection DR légèrement supérieur avec l'algorithme Σ - Δ , notre algorithme avec le nouveau filtre montre dans le tableau 2 une meilleure adaptabilité et moins de faux positifs. Cependant, le même algorithme avec estimation du fond montre au tableau 3 un faible taux de détection.



FIG. 6 : Séquences testées

Tab. 1 : Résultats avec algorithme Σ - Δ [7] (N=5) avec et sans fonction de détection de translation locale (F et NF)

| Séquence | Métriques (%) | | | | | |
|--------------|---------------|------|------|------|------|------|
| | FAR | | DR | | FPR | |
| | NF | F | NF | F | NF | F |
| Hall | 73,2 | 5,48 | 98,1 | 90,2 | 85,5 | 1,37 |
| kwbB | 80,7 | 14,9 | 96,2 | 88,4 | 64,9 | 2,15 |
| Walk | 91,2 | 87,3 | 98,7 | 98,6 | 64,4 | 41,5 |
| Pets 2002 | 77,1 | 52,3 | 95,1 | 90,7 | 65,4 | 18,2 |
| dtneu_schnee | 70,6 | 61,1 | 99,9 | 94,4 | 99,5 | 57,2 |

Tab. 2 : Résultats avec nouveau filtre (Δ_{n1})

| Séquence | Métriques (%) | | | | | |
|--------------|---------------|------|------|------|------|------|
| | FAR | | DR | | FPR | |
| | NF | F | NF | F | NF | F |
| Hall | 10,3 | 0,77 | 87,8 | 60,8 | 2,56 | 0,11 |
| kwbB | 15 | 6,48 | 85,3 | 77,3 | 2,07 | 0,71 |
| Walk | 80,7 | 58,2 | 99,2 | 96,9 | 26,8 | 8,32 |
| Pets 2002 | 54,4 | 3,59 | 89,2 | 71,7 | 18,8 | 0,43 |
| dtneu_schnee | 62,8 | 19,1 | 91,1 | 78,1 | 58,5 | 6,07 |

Tab. 3 : Résultats avec estimation du fond (Δ_{n2})

| Séquence | Métriques (%) | | | | | |
|--------------|---------------|------|------|------|------|------|
| | FAR | | DR | | FPR | |
| | NF | F | NF | F | NF | F |
| Hall | 14,8 | 1,16 | 76,8 | 47,7 | 3,34 | 0,13 |
| kwbB | 19 | 6,25 | 79,4 | 69,2 | 2,51 | 0,6 |
| Walk | 69,8 | 46,4 | 97,4 | 94,3 | 14,2 | 4,86 |
| Pets 2002 | 57,8 | 2,69 | 78,7 | 61,2 | 18,6 | 0,27 |
| dtneu_schnee | 65,1 | 17,4 | 89,5 | 67,7 | 63,7 | 4,58 |

A partir de simulations analogiques (SPICE), nous avons pu établir la puissance dévolue à chacun des éléments de l'architecture. A partir de ces informations nous avons établi que la puissance consommée est de 40mW sur des images de 240×320 , à 25 images.s⁻¹. Cette consommation est du même ordre de grandeur que celle d'un capteur CMOS classique faisant l'acquisition d'une image pleine résolution mais sans processeur pour effectuer des traitements équivalents. A titre de

comparaison, un traitement SIMD pleine résolution d'une matrice de 128×128 pixels [12] induit une consommation de 240 mW tandis que le portage sur processeur numérique requiert une puissance de l'ordre du Watt.

5. Conclusion

Un algorithme nécessitant peu de ressources de calcul et de mémoire réalisant une segmentation de mouvement sur des séquences pleine résolution, destiné à une implémentation sur un capteur d'image CMOS intelligent et faible consommation, a été présenté. Une robustesse aux différentes contraintes extérieures et intérieures a été exposée, de même que la faible complexité et consommation de l'algorithme.

Références

- [1] Moutault S., Mathias H., Klein J.-O., Dupret A., "An improved analog computation cell for Paris II, a programmable vision chip", Circuits and Systems, 2004. ISCAS '04, Volume 5, 23-26 May 2004 Page(s):V-453 - V-456 Vol.5.
- [2] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, Vol.34, N°3, 10/2004.
- [3] Joo S., Zheng Q., "A temporal variance-based moving target detector", IEEE Workshop PETS, Jan. 2005.
- [4] Abdelkader M.F., Chellappa R., Qinfen Zheng, Chan A.L., "Integrated Motion Detection and Tracking for Visual Surveillance", ICVS '06, IEEE.
- [5] Vazquez J.F. and al., "Adaptive threshold for motion detection in outdoor environment using computer vision", ISIE 2005, IEEE, 20-23 June 2005, Volume: 3, Page(s): 1233- 1237 vol. 3.
- [6] Richefeu J., Manzanera A., "Motion Detection with smart sensor", 9th Congress Young Searchers in Computer Vision - ORASIS 05, May 05.
- [7] Manzanera A., Richefeu J., "A robust and computationally motion detection algorithm based on background estimation", ICVGIP'04, December 2004.
- [8] Klein J.-O., Lacassagne L., Mathias H., Moutault S., Dupret A., "Low power image processing: analog versus digital comparison", CAMP 2005.
- [9] L. Lacassagne, M. Milgram, and P. Garda., "Motion detection, labeling, data association and tracking in real-time on risc computer", In Proc. IEEE ICIAP, pages 520.525, 1999.
- [10] J. Denoulet, G. Mostafaoui, L. Lacassagne, A. Mériçot, "Implementing Motion Markov Detection on General Purpose Processor and Associative Mesh", IEEE, CAMP 2005, 4-6 July 2005, Palermo, Italy.
- [11] Black J., Ellis T.J., and Rosin P., "A novel method for video tracking performance evaluation", IEEE Workshop VS-PETS, pages 125-132, October 2003.
- [12] P. Dudek and S.J. Carey, "General-purpose 128×128 SIMD processor array with integrated image sensor", Electronic letters, 8th June 2006 Vol. 42 No. 12.