Algorithme de détection de mouvement par modélisation markovienne Mise en oeuvre sur DSP

MRF-based Motion Detection Algorithm Image Processing Board Implementation

par Alice CAPLIER, Christophe DUMONTIER, Franck LUTHON, Pierre-Yves COULON

Laboratoire de Traitement d'Images et Reconnaissance de Formes
Institut National Polytechnique de Grenoble
LTIRF, INPG, 46 avenue Félix-Viallet
38031 Grenoble Cedex, France
Tel : (33) 76 57 43 72 Fax : (33) 76 57 47 90
Email : luthon@tirf.inpg.fr

résumé et mots clés

Dans un premier temps, nous présentons un algorithme de détection de mouvement dans les séquences d'images acquises avec une caméra fixe (étiquetage binaire de l'image en pixels fixes ou mobiles). L'approche est basée sur une modélisation des interactions spatio-temporelles entre étiquettes par un champ de Markov faisant intervenir trois images successives. Cet algorithme se caractérise par sa robustesse, sa rapidité de convergence et sa simplicité (limitation du nombre d'heuristiques). Ensuite, on montre comment le modèle proposé s'étend aisément au cas du multi-étiquetage moyennant une modification de la stratégie d'initialisation des champs d'étiquettes et de la stratégie de relaxation. Ceci permet de faire non plus simplement une détection binaire des zones mobiles mais une détection multi-étiquette non supervisée (discrimination des divers objets mobiles et estimation de leur nombre) et un suivi court-terme des objets mobiles. Enfin, nous présentons les premiers résultats d'une mise en oeuvre matérielle de l'algorithme de détection binaire sur une carte générique de traitement d'images à base d'un DSP.

Détection de mouvement, Séquence d'images, Champ de Markov, Multi-étiquetage, Implantation matérielle, DSP.

abstract and key words

First, we present a motion detection algorithm for image sequences acquired with a static camera (binary labelling of each pixel according to static or mobile areas). The approach is based on a Markov Random Field modelling of the spatiotemporal interactions between labels. The algorithm works on three consecutive frames. Robustness, convergence speed and simplicity (few heuristics) are the main characteristics of the algorithm. Then, it is shown how the proposed model may be easily extended to the case of multilabelling by modifying the initialisation of the label field and the relaxation strategy. Instead of a bare binary detection of moving areas, multilabelling enables a discrimination between different moving objects, an estimation of their number and a short-term tracking of moving areas. Finally, we present the first results about a hardware implementation of the binary detection algorithm on a general purpose image processing board build around a DSP.

Motion detection, Image sequences, Markov Random Field (MRF), Multilabelling, Hardware Implementation, DSP, Image processing board.

1. introduction

L'analyse du mouvement présente un intérêt majeur en vision par ordinateur étant donné le champ des applications possibles (contrôle du trafic routier, guidage de robot, compression de séquences d'images, diagnostic médical . . .). Traditionnellement, cette analyse englobe quatre aspects principaux : détection de mouvement, estimation de flux optique, segmentation de mouvement et interprétation du mouvement [2]. L'approche markovienne quant à elle a été largement utilisée en traitement d'images

ces dix dernières années, initialement pour la segmentation de texture et la restauration d'images bruitées [7, 8]. Plus récemment, elle a été utilisée en analyse de séquences d'images [17, 9]. Des modèles de détection de mouvement basés sur la modélisation markovienne ont déjà été proposés, en particulier dans [3].

Dans cet article, nous présentons un algorithme de détection de mouvement dérivé de celui présenté dans [3]. La détection de mouvement, dans le cadre de l'analyse de séquences d'images acquises par une caméra fixe, est abordée comme un problème d'étiquetage binaire des pixels de chaque image : il s'agit de faire la distinction entre les pixels appartenant à une zone statique et ceux appartenant à une zone mobile. Les interactions spatiales et temporelles entre étiquettes voisines de la séquence sont modélisées par un champ de Markov qui représente le modèle *a priori*. La solution correspond à la configuration la plus probable d'un champ de primitives (*étiquettes*) étant donné un champ de données (*observations*). Le problème se ramène en pratique à la recherche du minimum d'une fonction d'énergie à l'aide d'un algorithme de relaxation déterministe. Les modifications par rapport à l'algorithme original présenté dans [3] consistent à simplifier le modèle en limitant au maximum les heuristiques dans le but d'une part, d'adapter l'algorithme dans un cadre de détection multi-étiquette et d'autre part, de permettre une mise en oeuvre "temps réel". Des tests sur des séquences synthétiques et naturelles ont permis d'étudier les performances de l'algorithme modifié. La qualité des résultats est tout à fait comparable à celle de l'algorithme initial : même qualité de reconstruction des masques des objets mobiles, même robustesse et même rapidité de convergence.

En revanche, les avantages de notre modèle de détection résident dans les deux points suivants :

• dans un cadre multi-étiquette, le modèle, associé à une initialisation multi-étiquette et une relaxation multi-voisinage, permet de distinguer les divers objets mobiles présents dans la scène et de réaliser un suivi court-terme (liaison temporelle des cartes de détection successives à partir de trois images). L'originalité de notre algorithme est que le multi-étiquetage est directement intégré au modèle, et non pas réalisé après coup sur le résultat de la détection binaire comme c'est le cas dans [13];

• la modélisation markovienne et la minimisation de la fonction d'énergie qui en découle, impliquent en programmation logicielle une charge de calcul conséquente même lorsqu'on utilise un algorithme de relaxation déterministe. Il importe donc de développer des modèles markoviens pouvant être mis en oeuvre sur du matériel afin d'atteindre des cadences de traitement "temps réel" et des systèmes peu encombrants. Les études que nous avons menées montrent la faisabilité d'implantation de notre algorithme de détection de mouvement sur une carte de traitement d'images à base de DSP (cette mise en oeuvre est développée dans la suite de l'article), sur une machine parallèle [5bis] et sur un réseau résistif analogique VLSI en technologie CMOS [5bis]. Avec l'algorithme initial [3], l'implantation, notamment sur un réseau analogique, est beaucoup trop complexe, voire impossible.

Dans un premier temps, nous détaillons les caractéristiques de l'algorithme de détection de mouvement (étiquetage binaire). Ensuite, nous donnons les modifications à apporter pour étendre ce modèle au cas d'une détection de mouvement multi-étiquette. Celles-ci concernent essentiellement la phase d'initialisation et la stratégie de relaxation. Nous proposons une initialisation partielle du champ courant avant relaxation à partir de la carte multi-étiquette passée et de la carte binaire des changements temporels présents puis, pour les points restant indécis, une relaxation multi-voisinage basée sur différentes initialisations virtuelles possibles de ces voisins indécis. Enfin, nous exposons la mise en oeuvre de l'algorithme de détection binaire sur une carte générique de traitement d'image équipée d'un DSP ST18941 fonctionnant à une fréquence d'horloge de 10MHz. Après une présentation du matériel utilisé pour cette implantation, nous détaillons la mise en oeuvre logicielle et les temps de calcul nécessaires à chaque étape de la détection de mouvement.

2. détection de mouvement par modélisation markovienne

La modélisation markovienne associée à l'estimation bayésienne est un outil statistique intéressant en traitement d'images car elle permet d'intégrer dans un même modèle des informations de nature différente et de réaliser une régularisation des solutions trouvées, en spécifiant simplement l'*a priori* du modèle par l'intermédiaire de potentiels énergétiques.

Nous adopterons les notations suivantes :

• E et O représentent les variables aléatoires associées respectivement au champ des étiquettes et au champ des observations à l'instant t;

• e et o représentent une réalisation particulière de E et O respectivement;

• $e(s)$ et $o(s)$ représentent la valeur des champs e et o au pixel ou site s de coordonnées (x, y);

• S représente l'ensemble des sites d'une image.

A chaque fois que l'instant considéré sera différent de l'instant courant t, un indice temporel supplémentaire sera ajouté dans ces notations.

2.1. observations et primitives

Moyennant les hypothèses de **caméra fixe** et **d'éclairement quasi constant de la scène**, il existe un lien entre objets mobiles et changements temporels de la fonction de luminance. Cela conduit

naturellement à prendre comme observation la valeur absolue de la dérivée temporelle de la fonction de luminance $I(x, y, t)$ qui est approchée numériquement par une différence entre les instants t et $t - dt$:

$$o(s) = |I(x, y, t) - I(x, y, t - dt)| \quad (1)$$

Par ailleurs, les étiquettes pertinentes dans le cas de la détection sont les suivantes :

- $e(s) = a$ si le pixel appartient à un objet mobile,
- $e(s) = b$ si le pixel appartient au fond fixe.

On suppose que le champ des étiquettes suit la propriété de Markov relativement au voisinage spatio-temporel V de la figure 1 :

$$P[E(s) = e(s) \ / \ E(r) = e(r), r \neq s, r \in S]$$
$$= P[E(s) = e(s) \ / \ E(r) = e(r), r \in V]$$

où $P[X]$ dénote la probabilité de l'événement X. Le choix d'une étiquette en un site s ne dépend donc que des étiquettes des points voisins de s. Le voisinage choisi est supposé contenir toutes les informations utiles pour prendre une décision.

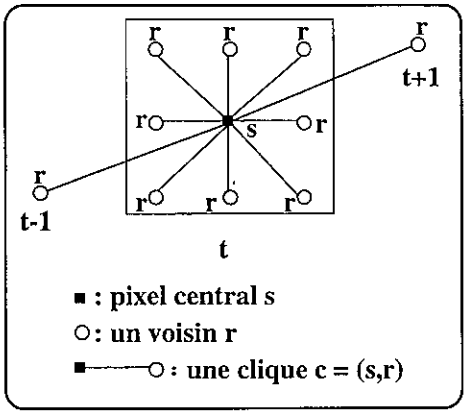

Figure 1. – Voisinage spatio-temporel et cliques binaires.

2.2. critère d'estimation et fonction d'énergie

Le champ des étiquettes est estimé au sens du critère du Maximum A Posteriori (MAP). Il conduit à la recherche de la configuration la plus probable du champ d'étiquettes par maximisation de la probabilité conditionnelle des étiquettes relativement aux observations [8] :

$$\max_e P[E = e/O = o]$$

D'après le théorème de Bayes, on a la relation :

$$P[E = e/O = o] = \frac{P[O = o/E = e]P[E = e]}{P[O = o]}$$

$P[O = o]$ est une constante vis-à-vis de la maximisation à réaliser car les observations sont des données du problème. D'après le théorème de Hammersley-Clifford (équivalence entre champ de Markov et distribution de Gibbs), la probabilité *a priori* des étiquettes s'exprime à partir d'une fonction d'énergie de modèle U_m et d'une constante de normalisation Z [8]

$$P[E = e] = \frac{1}{Z} \exp(-U_m)$$

Relativement aux cliques de la figure 1, la forme générique de la fonction d'énergie est :

$$U_m = \sum_{s \in S} u_m(e(s)) \quad (2)$$

$$\text{avec} \quad u_m(e(s)) = \sum_{c \in C_s} V_c(e(s), e(r))$$

où $V_c(e(s), e(r))$ est un potentiel élémentaire associé à la clique $c = (s, r)$ et où C_s est l'ensemble des cliques binaires associées au voisinage du pixel considéré. $u_m(e(s))$ représente donc l'énergie locale du modèle au pixel s affecté de l'étiquette $e(s)$. Pour conférer au modèle des propriétés de continuité spatiale et temporelle, on utilise des potentiels à niveau peu coûteux en temps de calcul :

$$V_c(e(s), e(r)) = \begin{cases} -\beta & \text{si } e(s) = e(r) \\ +\beta & \text{si } e(s) \neq e(r) \end{cases} \quad (3)$$

où le paramètre positif β dépend de la nature de la clique considérée. Nous définissons un paramètre β_s pour les cliques spatiales, un paramètre β_p pour les cliques temporelles passées et un paramètre β_f pour les cliques temporelles futures. Nous introduisons une anisotropie entre le passé et le futur : en pratique, nous choisissons $\beta_f > \beta_p$ afin de traiter le problème des discontinuités de mouvement en favorisant l'innovation apportée par le futur. Cet avantage donné au futur permet en particulier une bonne élimination de la zone d'écho (zone de fond découverte lors du mouvement). En effet, dans ce cas de figure, le voisin temporel passé porte l'étiquette de pixel mobile a et le voisin temporel futur porte celle de pixel fixe b. Or c'est l'étiquette fixe b qu'il faut choisir pour éliminer rapidement l'écho.

Quant à la probabilité conditionnelle des observations relativement aux étiquettes $P[O = o/E = e]$, elle résulte d'une relation entre observations et étiquettes [3] :

$$o(s) = \Psi(e(s)) + n \text{ avec } \Psi(e(s)) = \begin{cases} 0 & \text{si } e(s) = b \\ \alpha > 0 & \text{sinon} \end{cases} \quad (4)$$

où n est un bruit gaussien centré de variance σ^2. La fonction Ψ modélise les observations. En effet, si le pixel appartient à un objet fixe, il n'y a pas de changement temporel de la fonction de luminance, donc l'observation est quasi nulle. En revanche, si le pixel appartient à un objet mobile, il y a changement temporel et on suppose que l'observation est proche d'une valeur

α, correspondant approximativement à la valeur moyenne des observations non nulles.

A partir de cette relation, l'opposé du logarithme de la probabilité conditionnelle des observations relativement aux étiquettes s'exprime comme une énergie d'adéquation U_a :

$$U_a = \sum_{s \in S} u_a(e(s)) \text{ avec } u_a(e(s)) = \frac{1}{2\sigma^2}[o(s) - \Psi(e(s))]^2 \quad (5)$$

$u_a(e(s))$ est l'énergie locale d'adéquation au pixel s.

Finalement, la maximisation de la probabilité des étiquettes étant donné les observations est équivalente à la minimisation d'une fonction d'énergie totale $U = U_m + U_a$ constituée des deux termes définis précédemment :

- U_m, énergie du modèle *a priori*, assure la régularisation de la solution;
- U_a, énergie d'adéquation, assure une bonne cohérence de la solution par rapport aux données observées (attache aux données).

2.3. relaxation et algorithme de détection

La minimisation de la fonction d'énergie est un problème non trivial car cette fonction est a priori non convexe. Vu notre objectif de mise en oeuvre temps réel, nous avons d'emblée écarté les algorithmes de relaxation stochastique tels le recuit simulé [8] et nous nous sommes tournés vers l'ICM (*Iterated Conditional Modes*), algorithme de relaxation déterministe [1]. Cet algorithme est sous-optimal puisque, n'autorisant un changement d'étiquette que si celui-ci engendre une diminution de la fonction d'énergie, il ne garantit pas de trouver le minimum global. Cependant, si l'initialisation est suffisamment soignée, cet algorithme conduit à des résultats satisfaisants.

La figure 2 donne l'organigramme de l'algorithme de détection de mouvement. A un instant t, l'algorithme travaille à partir de 3 images successives de la séquence. Le champ passé E_{t-1} résulte de la relaxation à l'instant précédent. La première étape consiste à initialiser les champs présent $\hat{E}_t$ et futur $\hat{E}_{t+1}$ puisque nous travaillons sur un voisinage temporel symétrique (cf. figure 1). En pratique, ces initialisations $\hat{E}_t$, $\hat{E}_{t+1}$ sont issues respectivement des champs d'observations O_t et O_{t+1} binarisés par utilisation d'un test de maximum de vraisemblance [10]. Certes, le champ $\hat{E}_{t+1}$ ainsi obtenu est grossier, mais il s'avère qu'il contient cependant l'information pertinente (zones de l'image ayant subi une transition fond-objet ou objet-fond) aidant à la localisation des frontières de mouvement. Ensuite, une relaxation spatiale intégrant des informations spatio-temporelles est effectuée sur le champ courant E_t. Pour chaque site de ce champ, les deux étiquettes possibles $\{a, b\}$ sont testées et l'étiquette qui induit

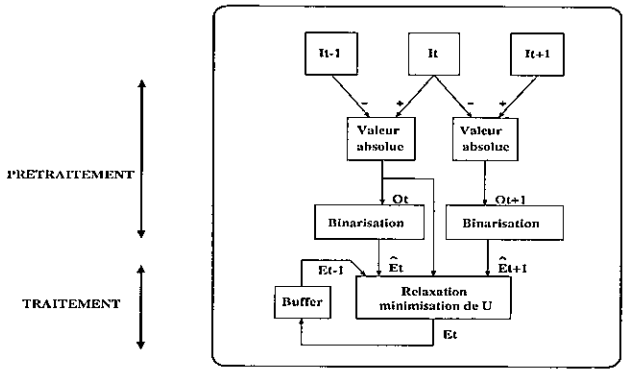

Figure 2. – Organigramme pour la détection binaire.

la plus grosse diminution de la fonction d'énergie locale est conservée. Ce processus est itéré[1] jusqu'à convergence.

2.4. résultats et performances

Une série de tests sur diverses séquences a permis de mettre en évidence les points suivants :

- l'algorithme est robuste : les mêmes valeurs de paramètres restent valables pour toutes les séquences que nous avons traitées : $\beta_s = 20$, $\beta_p = 10$, $\beta_f = 30$ et $\alpha = 20$;
- le nombre d'itérations nécessaire pour atteindre la convergence est peu élevé (moins d'une dizaine d'itérations dans la majorité des cas). Il varie évidemment en fonction de l'amplitude du mouvement entre deux images. Une étude plus approfondie montre que, dans le cas de séquences pour lesquelles les déplacements restent modérés par rapport à la cadence d'acquisition des images, 4 itérations sont suffisantes, les itérations suivantes nécessaires pour atteindre le critère théorique de convergence [2] n'apportant que très peu de modifications quant à la valeur de l'énergie finale. Voilà pourquoi dans la mise en oeuvre temps réel, le critère de convergence porte sur un nombre prédéfini d'itérations; en l'occurrence, nous nous sommes limités à **4 itérations**.

La figure 3 présente un exemple de détection de mouvement dans le cas d'une séquence synthétique contenant un cercle non uniformément éclairé qui se dilate et un carré qui se translate de gauche à droite avec un déplacement de 3 pixels par image. Cette figure présente les champs d'étiquettes initiaux issus de la binarisation des observations, les masques finaux (zones d'étiquettes mobiles) après relaxation et la superposition des contours des masques sur la séquence originale. La relaxation du modèle markovien a joué son rôle d'homogénéisation spatio-temporelle des masques.

1. Une itération correspond à un balayage complet de l'image.
2. Un balayage complet d'image sans modification d'étiquettes.

En haut de la figure 4, la séquence originale *Trevor* est présentée, la cadence des images étant de 15 images par seconde [3]. Au milieu, nous présentons les résultats de la détection de mouvement. Le manque d'information de mouvement sur les zones très homogènes de l'image telles que les mains ou le coude (observations quasi nulles) a une répercussion sur la qualité des résultats. De plus, dans ce cas précis, les zones en question correspondent à des régions où le mouvement est plus faible : les mains étant posées sur la table, elles bougent très peu par rapport au reste du corps. En bas, on présente la superposition des contours des masques trouvés sur la séquence d'images pour mettre en évidence la qualité de la détection. Nous constatons que la précision au niveau des contours n'est pas parfaite : cela est dû au fait que, dans le modèle considéré, la prise de décision est relative à l'information de mouvement contenue dans le voisinage de la figure 1 et ne fait intervenir aucune information relative aux contours des objets.

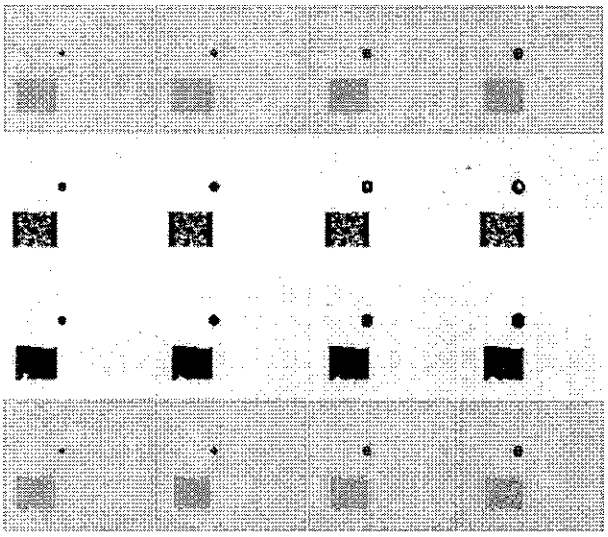

Figure 3. – Détection de mouvement (les pixels mobiles sont représentés en noir et les pixels fixes en gris très clair) : de haut en bas : 1) séquence d'images synthétiques; 2) initialisation des champs d'étiquettes (observations binarisées); 3) masques des objets mobiles après relaxation; 4) superposition des contours des masques sur la séquence d'images.

3. extension à la détection multi-étiquette

Dans ce paragraphe, nous exploitons le premier avantage de notre algorithme de détection : la possibilité d'étendre le modèle au cas du multi-étiquetage (toujours dans le cadre de l'analyse de séquences d'images acquises avec une caméra fixe). Nous expliquons quelles sont les modifications à apporter pour pouvoir non seulement détecter mais aussi discriminer entre eux divers objets mobiles. L'intégration d'information temporelle sur trois images permet un suivi court-terme de ces différents objets en mouvement (lien temporel entre les cartes de détection multi-étiquette successives).

3.1. modèle multi-étiquette et relaxation

Notre objectif est désormais d'attribuer une étiquette discriminante pour chaque objet mobile différent. Cependant, notre algorithme ne prétend pas traiter tous les cas de figure. En particulier, il ne permet pas de faire la distinction, après leur séparation, de deux objets initialement connexes. Il ne permet pas non plus de résoudre le problème difficile des occultations puisque l'intégration temporelle ne porte que sur trois images. Mais il est malgré tout possible de traiter le cas de l'occultation temporaire d'un objet par le fond fixe en gardant une mémoire des événements passés au moyen d'un filtre de Kalman (traitement qui ne sera pas détaillé ici) [5bis].

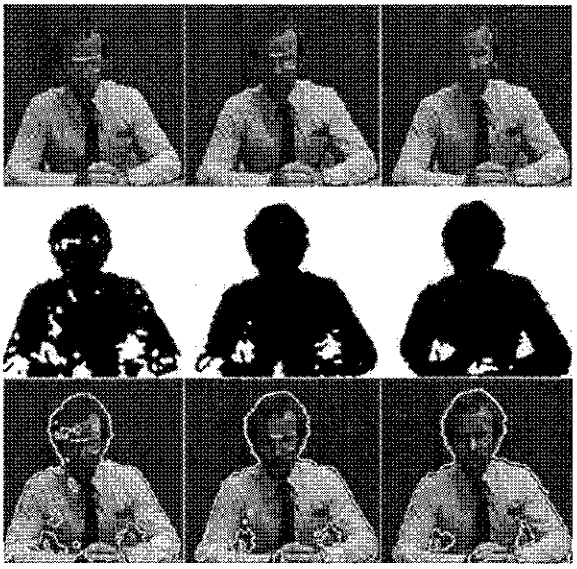

Figure 4. – Détection de mouvement (les pixels mobiles sont représentés en noir et les pixels fixes en gris très clair) : de haut en bas : 1) séquence *Trevor*; 2) masques issus de la détection de mouvement; 3) superposition des contours des masques sur la séquence d'images.

L'ensemble des étiquettes possibles à un instant t est à présent étendu :

• $e(s) = a_i$ si le pixel appartient au $i^{\text{ème}}$ objet mobile, $i \in \{1...M_t\}$

3. Nous avons considéré une image sur deux afin d'éviter le problème des mouvements subpixels, un déplacement d'une fraction de pixel entre 2 images ne pouvant pas être pris en compte dans les observations. Le cas des déplacements subpixels peut être avantageusement traité dans un cadre multi-résolution [4].

• $e(s) = b$ si le pixel est fixe,

• $e(s) = a_0$ si le pixel appartient à un nouvel objet mobile [4].

où M_t représente **le nombre total (*a priori* inconnu) d'objets en mouvement** à l'instant courant t. Ce nombre est réévalué à chaque instant. La détection multi-étiquette est non supervisée; aucune information *a priori* n'est nécessaire sur le nombre total d'objets présents dans la scène analysée. Nous avons ajouté l'étiquette a_0 afin de pouvoir gérer le cas de l'apparition de nouveaux objets. De ce fait, on réalise conjointement la détection multi-étiquette et l'estimation du nombre d'objets présents.

Les observations utilisées pour la détection multi-étiquette sont les mêmes que pour la détection binaire (cf. équation (1)).

Les deux fonctions d'énergie définies précédemment ne nécessitent pas non plus de modification pour être utilisées dans le cadre multi-étiquette. L'énergie du modèle *a priori* U_m, définie à partir d'une somme de potentiels élémentaires ne faisant intervenir que des tests d'égalité ou d'inégalité entre étiquettes, n'a pas besoin d'être modifiée (équations (2) et (3)). De même, la fonction Ψ modélisant les observations, ainsi que l'énergie d'adéquation qui en découle peuvent être utilisées telles quelles dans le cadre multi-étiquette (équations (4) et (5)).

Le schéma général de relaxation à partir de l'algorithme des ICM est conservé (cf. figure 5). Les modifications nécessaires à la gestion du multi-étiquetage sont matérialisées par les zones en pointillés sur la figure 5.

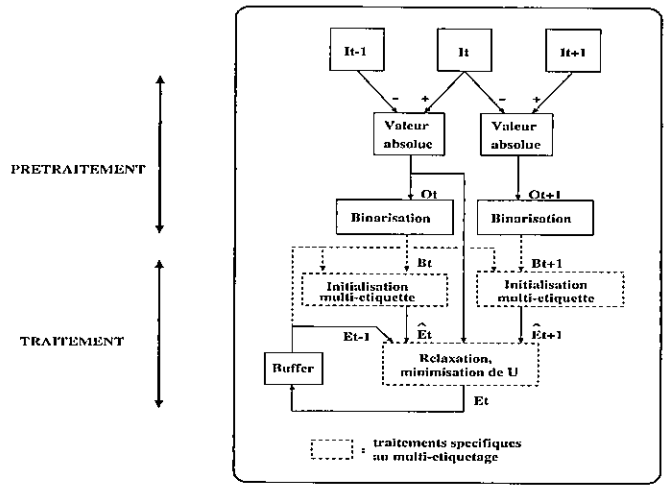

Figure 5. – Organigramme pour la détection multi-étiquette.

4. A l'issue de la phase d'initialisation, l'étiquette a_0 correspond temporairement à une étiquette de *mouvement indéterminé*, comme on l'explicite au paragraphe 3.2

3.2. initialisation et relaxation

Les modifications majeures concernent la stratégie d'initialisation des champs présent et futur. Nous avons désormais besoin d'une initialisation multi-étiquette de ces champs et cette initialisation doit être aussi soignée que possible puisque l'algorithme de relaxation utilisé (ICM) y est sensible.

Nous proposons une initialisation partielle suivie d'une relaxation multi-voisinage.

3.2.1. *initialisation partielle*

Nous nous intéressons d'abord à l'initialisation du champ présent. Elle est réalisée à l'aide du champ passé multi-étiquette E_{t-1} et du champ binaire présent B_t issu du champ des observations O_t (par utilisation d'un test de maximum de vraisemblance). B_t contient tous les points où il y a eu un changement temporel significatif de la fonction de luminance. A l'instant t, pour chaque pixel s détecté en changement temporel dans B_t, nous testons l'étiquette $e_{t-1}(s)$ du pixel correspondant dans le champ précédent :

• si $(\exists i \in \{1, \ldots M_{t-1}\} \ / \ e_{t-1}(s) = a_i)$, le pixel est initialisé avec la même étiquette mobile a_i;

• si $(e_{t-1}(s) = b)$, aucune information n'est disponible dans le passé pour initialiser ce pixel. Nous attribuons alors à un tel pixel l'étiquette initiale temporaire a_0 correspondant à un *mouvement indéterminé*. En effet, le pixel est mobile puisque détecté en changement temporel, mais aucune information dans le passé ne permet de lui attribuer une étiquette initiale fiable.

La figure 6 donne une illustration du résultat de cette phase d'initialiation.

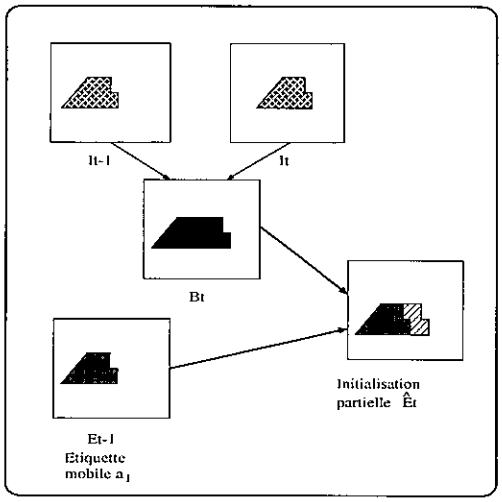

Figure 6. – Initialisation partielle : la zone hachurée correspond aux pixels dont l'initialisation est indéterminée (pixels avec l'étiquette a_0) et la zone grise correspond aux pixels dont l'initialisation est définie (étiquette a_1 provenant du champ passé E_{t-1})

L'initialisation multi-étiquette du champ futur est obtenue de la même manière. L'utilisation du même champ passé E_{t-1} (le seul dont on dispose effectivement) pour initialiser le futur conduit évidemment à un nombre de points indécis plus important pour $\hat{E}_{t+1}$ que pour $\hat{E}_t$, mais les résultats obtenus avec une telle initialisation s'avèrent satisfaisants.

3.2.2. Relaxation multi-voisinage

Sélection des voisinages virtuels. A l'issue de la phase d'initialisation partielle, nous disposons d'un champ initial pour lequel certains points portent l'étiquette temporaire a_0 de *mouvement indéterminé*. Or, lors de la relaxation, le calcul de l'énergie de modèle $u_m(e(s))$ fait intervenir les étiquettes des points voisins. Il est donc indispensable que tous les voisins soient spécifiés, c'est-à-dire que leur étiquette soit déterminée. La spécification des voisins indécis va être réalisée de manière virtuelle en cours de relaxation. La figure 7 présente un schéma permettant d'illustrer la manière dont cette spécification est effectuée. Soit le pixel central s dont le voisinage spatio-temporel contient des pixels étiquetés a_0. Puisque les voisins indécis sont des pixels détectés en changement temporel, une bonne étiquette pour ces voisins est soit l'une des étiquettes mobiles a_i déjà présentes dans le voisinage, soit l'étiquette a_0 indiquant désormais la présence d'un nouvel objet et non plus un mouvement indéterminé comme c'était le cas dans la phase d'initialisation. L'examen du voisinage spatio-temporel de s fournit l'ensemble des étiquettes virtuelles possibles pour les voisins indéterminés (en l'occurrence une seule étiquette mobile possible a_1 dans le cas particulier de la figure 7). Nous limitons la recherche des étiquettes possibles au voisinage spatio-temporel du pixel s car nous supposons que ce voisinage est pertinent. A ces spécifications possibles issues de l'examen du voisinage, on ajoute de façon systématique une spécification virtuelle des pixels indéterminés par l'étiquette a_0 elle-même afin de traiter l'apparition éventuelle d'un nouvel objet. Cette démarche fournit un ensemble de voisinages virtuels possibles pour le pixel central s (dans le cas de notre schéma, il en résulte deux voisinages V_0 et V_1).

Notons qu'au cours de cette phase de spécification virtuelle, on n'attribue pas de façon définitive d'étiquette effective nouvelle aux voisins indécis; il ne s'agit que d'une prise de décision temporaire pendant le traitement du pixel courant s. Les étiquettes présentes dans le voisinage spatio-temporel du pixel courant sont recensées afin d'en déduire toutes les spécifications possibles des voisins indécis (en attribuant successivement la même étiquette virtuelle parmi toutes celles possibles à tous les pixels indécis de ce voisinage). Evidemmment, les voisinages virtuels ainsi définis peuvent être erronés dans le cas de deux objets connexes. Mais comme nous le montrons sur les résultats de la figure 9, les conséquences en sont minimes sur la qualité de la précision des contours de deux objets connexes.

Relaxation sur les voisinages virtuels. L'ensemble des étiquettes possibles pour le pixel courant s pendant la relaxation est $\{a_i, b, a_0\}, i \in \{1...M_{t-1}\}$. Lors de la première itération, nous calculons l'énergie locale du pixel courant pour chaque étiquette possible a_i, b ou a_0 relativement à tous les voisinages virtuels définis dans l'étape précédente (2 voisinages spatio-temporels V_1 et V_0 sur l'exemple de la figure 7). Le pixel courant reçoit l'étiquette qui produit l'énergie locale la plus faible, tous voisinages virtuels confondus. On recherche et on retient donc l'étiquette qui fournit **le minimum minimorum des énergies locales** quel que soit le voisinage virtuel considéré pour faire ce calcul d'énergie. Dans le cas de la figure 7, ce sera l'étiquette a_1. Remarquons toutefois qu'aucune décision définitive n'est prise relativement aux pixels du voisinage lors du traitement du pixel central s.

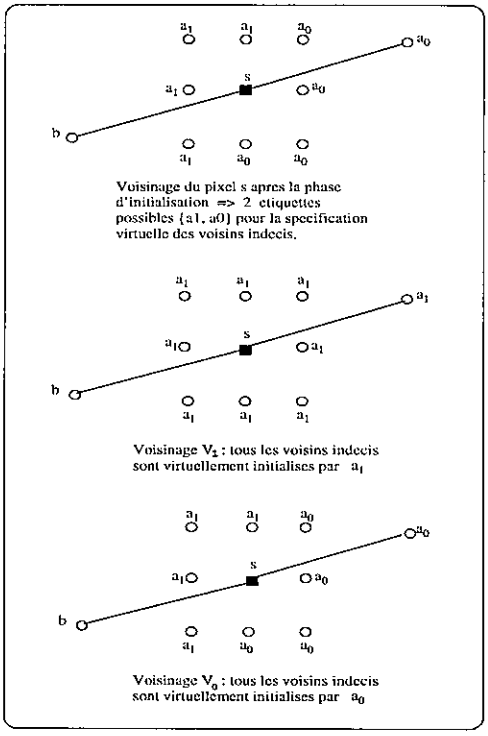

Figure 7. – Définition des voisinages virtuels : cas d'un pixel s à la frontière entre des pixels indécis (étiquette a_0) et des pixels initialisés (étiquette a_1). La dimension temporelle est représentée par le trait liant passé, présent et futur.

Dans le cas où c'est l'étiquette a_0 qui est conservée, elle indique la présence de nouveaux objets dans la scène. De nouvelles étiquettes mobiles $a_j, j > M_{t-1}$ sont attribuées par étiquetage en composantes connexes des zones portant l'étiquette a_0. De cette manière, il est possible d'estimer conjointement le nombre d'objets mobiles présents dans la scène à chaque instant. Ce nombre est alors remis à jour et devient M_t. A l'opposé, dans le cas de la disparition d'un objet de la scène, son étiquette est libérée et pourra éventuellement être réutilisée par la suite.

A cette relaxation multi-voisinage, il est impératif d'associer une stratégie de visite de sites orientée. En effet, sachant que la politique de visite des sites peut avoir une influence sur le résultat de l'ICM, nous traitons en premier les pixels les plus fiables,

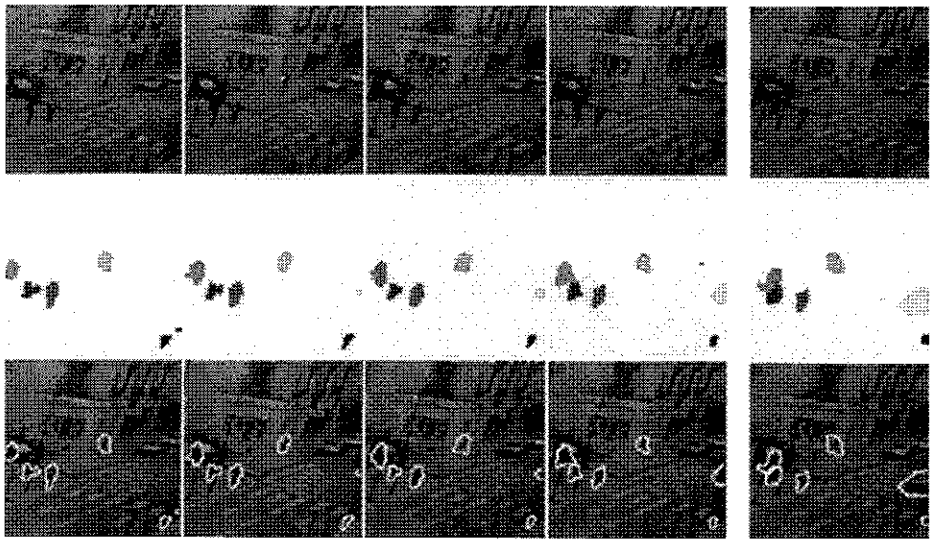

Figure 8. – Détection multi-étiquette (les pixels mobiles sont représentés en niveau de gris et les pixels fixes quasiment en blanc) : de haut en bas : 1) séquence d'images ($t = 3, 4, 5, 6$ et 12); 2) résultat de la détection multi-étiquette où chaque étiquette mobile est représentée par un niveau de gris différent; 3) superposition des contours des masques sur la séquence d'images.

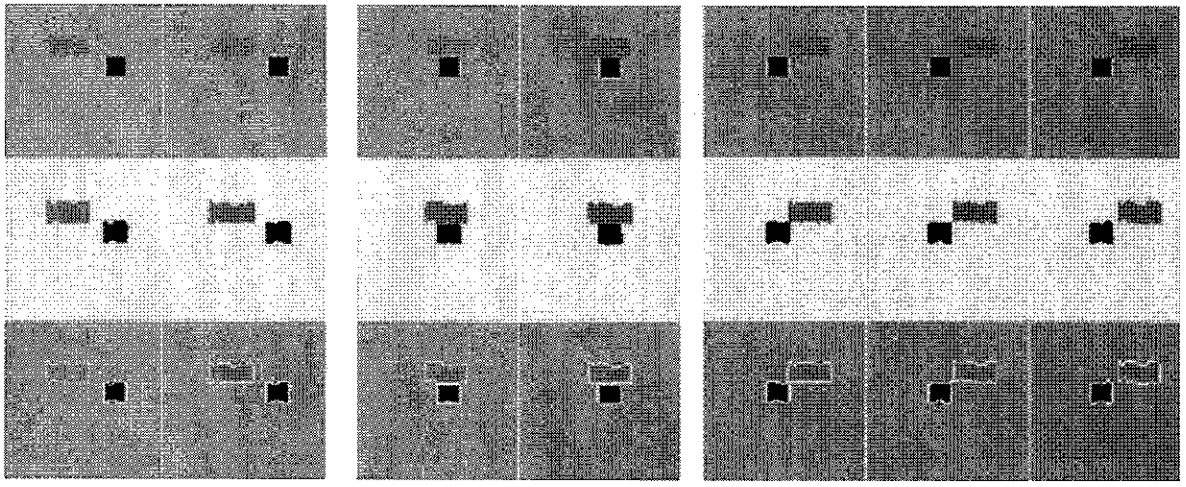

Figure 9. – De haut en bas : 1) séquence d'images synthétiques ($t = 3, 4, 15, 16, 23, 24, 25$); 2) résultat du multi-étiquetage où chaque étiquette mobile est représentée par un niveau de gris différent; 3) superposition des contours des masques sur la séquence d'images.

à savoir ceux dont le voisinage spatio-temporel ne contient pas de pixels indéterminés. Nous nous inspirons ici du principe de l'algorithme High Confidence First [6], en ce seul sens que c'est le degré de confiance qui oriente la visite des sites.

3.3. résultats

La figure 8 montre les résultats de la détection multi-étiquette obtenus pour une séquence d'images de taille 128×128 codées sur 8 bits. La scène a été filmée dans la rue, les objets mobiles sont donc essentiellement des piétons et des véhicules. La détection multi-étiquette fournit un ensemble de masques, chaque masque étant associé à une étiquette différente. L'initialisation obtenue à partir des masques du champ passé permet une liaison temporelle des étiquettes successives. Evidemment, cette liaison n'a pas lieu s'il n'y a pas de recouvrement entre les positions successives d'un même objet. En cas d'intersection vide entre les positions d'un objet aux instants $t-1$ et t, aucun des pixels du masque de cet objet n'a de voisin passé mobile. A l'issue de la première initialisation, tous les points du masque seront indécis et lors de la relaxation, c'est l'étiquette a_0 de nouvel objet qui sera conservée. Ce problème ne se pose pas tant que les déplacements restent faibles par rapport à la cadence d'acquisition des images.

La séquence d'images présentée est intéressante à double titre.

D'une part, elle permet d'illustrer le comportement de l'algorithme en cas d'apparition d'un nouvel objet. Les quatre premières images présentées sont consécutives. A $t = 4$, il apparaît une voiture tout à fait à droite. Lors de la relaxation, aucune des étiquettes mobiles déjà présentes ne convenant, cette voiture conserve l'étiquette a_0 de nouvel objet au cours des itérations successives si bien qu'à la fin, une nouvelle étiquette mobile lui est attribuée. L'estimation du nombre d'objets mobiles présents est donc réalisée conjointement au multi-étiquetage. La dernière image présente le résultat de la détection à $t = 12$, lorsque la voiture est plus nettement dans le champ de vision de la caméra.

D'autre part, cette séquence illustre le comportement de notre méthode dans le cas du regroupement temporaire de deux objets mobiles. En effet, les deux piétons les plus à gauche vont à la rencontre l'un de l'autre. Lors de leur regroupement ($t = 6$), chaque objet conserve néanmoins sa propre étiquette. Afin d'étudier plus précisément le cas du regroupement temporaire de deux objets mobiles, la figure 9 présente une séquence synthétique où deux objets vont à la rencontre l'un de l'autre à la vitesse de 2 pixels par image. Avant la rencontre ($t = 3$ et 4), chacun des deux objets possède sa propre étiquette, qu'il conserve lors du regroupement ($t = 15$ et 16), et également après la séparation ($t = 23, 24$ et 25).

Notons que le comportement de l'algorithme n'est pas fiable en cas d'occultation partielle ou totale d'objets. Le traitement de tels cas nécessiterait une intégration temporelle plus longue (par filtrage de Kalman par exemple). Néanmoins, notre objectif est ici de mettre en évidence les capacités de l'algorithme qui, utilisant uniquement l'information rudimentaire que constitue la différence temporelle de la fonction de luminance, permet toutefois une interprétation riche de la scène analysée dans des cas de figure non triviaux. Il fournit le nombre d'objets mobiles présents ainsi que leur localisation. La poursuite du centre de gravité de chacun des masques conduit ensuite à une information rudimentaire de vitesse. Enfin, la liaison temporelle de ces centres de gravité permet d'élaborer la trajectoire de chaque objet [14]. L'intégration au sein du modèle d'informations plus riches rendrait certes l'algorithme plus robuste mais au détriment de la complexité de calcul. Par exemple, dans [9], la prise en compte conjointe d'informations de contours et du flux optique permet de traiter le problème des occultations. Cependant, l'obtention d'un flux optique de bonne qualité est un problème difficile, souvent coûteux en calcul et dont le résultat n'est pas toujours exploitable en vue d'une segmentation en régions de mouvement homogène (quel critère utiliser pour segmenter le flux optique?).

4. implantation sur DSP

L'utilisation de la modélisation markovienne en traitement d'images pour résoudre divers problèmes a mis en évidence l'intérêt théorique de cette approche. Mais l'inconvénient majeur de cette modélisation réside dans la lourdeur des calculs engendrés par la relaxation. Il en résulte des cadences de traitement faible, ce qui est surtout un problème dans le cadre de l'analyse de scènes dynamiques. Pour accélérer les calculs associés à un algorithme basé sur la modélisation markovienne, deux approches ont été envisagées : utilisation de machines multi-processeurs [5, 18, 16] ou implantation sur réseau résistif analogique [15, 11, 12]. Ces deux types de mises en oeuvre tendent à tirer profit du caractère parallèle des calculs engendrés par la modélisation markovienne.

Pour notre algorithme de détection binaire, la cadence de traitement atteinte sur une station de travail Sun SPARC-10 est d'environ $2s$ pour le traitement d'une image de taille 128×128 ce qui est évidemment trop lent dans un contexte d'applications temps réel. Nous avons donc étudié les possibilités de mise en oeuvre de cet algorithme sur du matériel spécialisé. Les solutions machine parallèle et réseau analogique ont été envisagées [5] mais nous ne les décrirons pas ici. Nous avons également choisi d'implanter l'algorithme de détection de mouvement sur une carte générique de traitement d'images à base de DSP. En effet, pour une application réelle, la cadence de traitement n'est pas le seul critère impératif. Il se pose également des contraintes de coût, d'encombrement et de rapidité de développement. Une mise en oeuvre sur DSP permet de réaliser un bon compromis entre toutes ces contraintes et de plus, elle permet d'évaluer la possibilité de réaliser une chaîne complète de traitement "temps réel" (de l'acquisition des images à la visualisation des masques) et d'expertiser les points sensibles de l'algorithme dans un contexte "temps réel". Soulignons que la solution DSP est intéressante pour l'algorithme de détection binaire qui n'est pas trop complexe (charge de calcul non rédhibitoire pour le DSP) mais elle devient inadaptée dès que la complexité de l'algorithme considéré augmente. Voilà pourquoi la mise en oeuvre décrite concerne uniquement la détection de mouvement binaire (2 étiquettes seulement). L'algorithme de détection multi- étiquette engendre trop de calcul pour obtenir une implantation suffisamment rapide avec le matériel actuellement utilisé.

4.1. présentation du matériel

4.1.1. Matériel utilisé

L'algorithme de détection de mouvement est développé sur une carte de traitement d'images au format PC, carte basée sur l'utilisation d'un DSP ST18941 de SGS-Thomson cadencé à 10MHz. La figure 10 donne le synoptique de cette carte commercialisée par la société Secad-SA (sous la référence VPC941) et dont les principales caractéristiques sont les suivantes :

- 6 plans mémoire vidéo 512 x 512 sur 8 bits (VRAM);
- 1 plan graphique vidéo 512 x 512 sur 4 bits (VRAM) [5];
- 1 plan mémoire 512 x 512 sur 16 bits (DRAM);

5. Pour incrustation éventuelle.

- 8k x 16 bits de RAM statique;
- 8k x 32 bits de RAM programme;
- 1 processeur de signal ST18941
- 1 connecteur ISA 16 bits

L'acquisition vidéo se fait en temps réel et la carte permet la digitalisation d'un signal vidéo au standard CCIR, soit en noir et blanc, soit en couleur (format RVB).

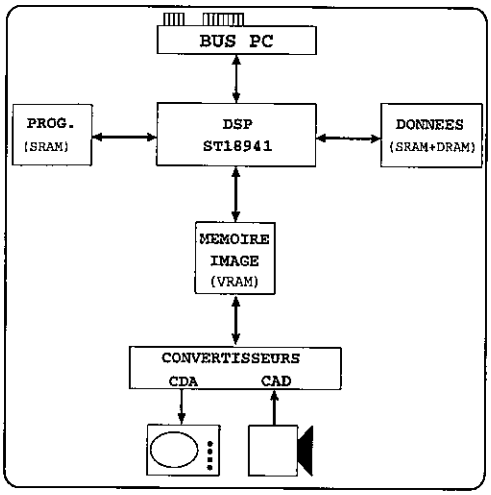

Figure 10. – Synoptique de la carte générique de traitement d'images.

4.1.2. *processeur de signal*

Le processeur de signal ST18941 possède une architecture Harvard, qui se caractérise par la séparation des données et des instructions, que ce soit pour les bus, ou pour les espaces mémoires.

Ce processeur possède 4 unités principales :

- l'unité arithmétique de données comprenant l'ALU, le multiplieur, le registre à décalage ainsi que les nombreux registres associés. Les calculs s'effectuent sur des nombres entiers de 16 ou 32 bits, réels ou complexes;
- l'unité de programme comprenant un contrôleur de programme (gestion des boucles, des branchements, des interruptions), ainsi qu'un espace mémoire programme. Les instructions, codées sur 32 bits, se décomposent en 4 champs (3 champs de gestion de données, 1 champ de calcul) et permettent une forte parallélisation des tâches;
- l'unité de données comprenant 3 blocs de RAM interne (XRAM, YRAM de 256 x 16 bits et CRAM de 128 x 16 bits) et 1 bloc externe (ERAM) réparti entre la RAM statique et la RAM dynamique (VRAM et DRAM);
- l'unité d'entrées/sorties comprenant le bus local (liaison DSP-ERAM, 16 bits d'adresses et de données), le bus d'instructions (16 bits d'adresses, 32 bits de données), le bus système (8 bits de données reliés au DSP par boîte aux lettres) et un port parallèle de 8 bits.

Le cycle instruction est de 100 ns. L'accès aux RAM internes et statiques nécessite un cycle d'instruction; l'accès aux RAM dynamiques nécessite soit 4 cycles d'instruction en mode aléatoire, soit 2 cycles d'instruction en mode page.

4.1.3. *conclusion*

De part son architecture spécifique, la carte VPC941 permet l'acquisition, la visualisation et le traitement d'images provenant d'une source vidéo standard. De plus, l'utilisation du processeur ST18941 offre la possibilité de traitements rapides (calculs entiers) sur les images acquises. En revanche, le calcul flottant n'est pas opérant sur ce processeur et la programmation doit se faire en assembleur car aucun compilateur C fiable et optimal n'est disponible. De plus, l'utilisation de RAM dynamique (mémoire vidéo ou DRAM) limite, par ses temps d'accès, la rapidité des calculs.

4.2. mise en oeuvre logicielle

La carte permet l'acquisition d'images couleurs de taille 512×512 en deux trames entrelacées. Mais l'information de mouvement étant contenue dans le signal de luminance, nous ne traitons que des images noir et blanc. De plus, les images traitées sont de taille réduite (128×128) afin d'éviter une charge de calcul trop importante pour la carte (générique) et le DSP de fréquence d'horloge 10MHz, tout en permettant l'obtention de résultats réalistes et interprétables (à terme, notre objectif est d'atteindre la demi cadence vidéo pour des images de taille 256×256). Les systèmes d'acquisition fournissant en général une image composée de deux trames entrelacées retardées de 20ms, le traitement ne porte que sur une seule trame (en l'occurrence la trame paire). En effet, entre deux trames, il n'y a pas d'information de mouvement suffisamment pertinente. De plus, la présence de deux plans vidéo (VRAM) et de mémoire dynamique de données (DRAM) permet de ne pas perturber les acquisitions d'images à cadence vidéo, tous les calculs étant effectués sur un masque stocké en DRAM.

La détection de mouvement peut se décomposer en deux parties (cf. figure 2) :

- un prétraitement comprenant le calcul des observations et le calcul des cartes binaires nécessaires à l'initialisation;
- la relaxation du champ de Markov (minimisation d'énergie) visant à obtenir les masques des différents objets mobiles.

4.2.1. *Prétraitement*

Le prétraitement suit le diagramme temporel de la figure 11.

Malgré des temps d'accès mémoire vidéo assez importants, il est possible d'effectuer le calcul des observations O_t à la cadence de 25 images/seconde. Par rapport à l'algorithme initial, la binarisation des observations est effectuée par simple comparaison par

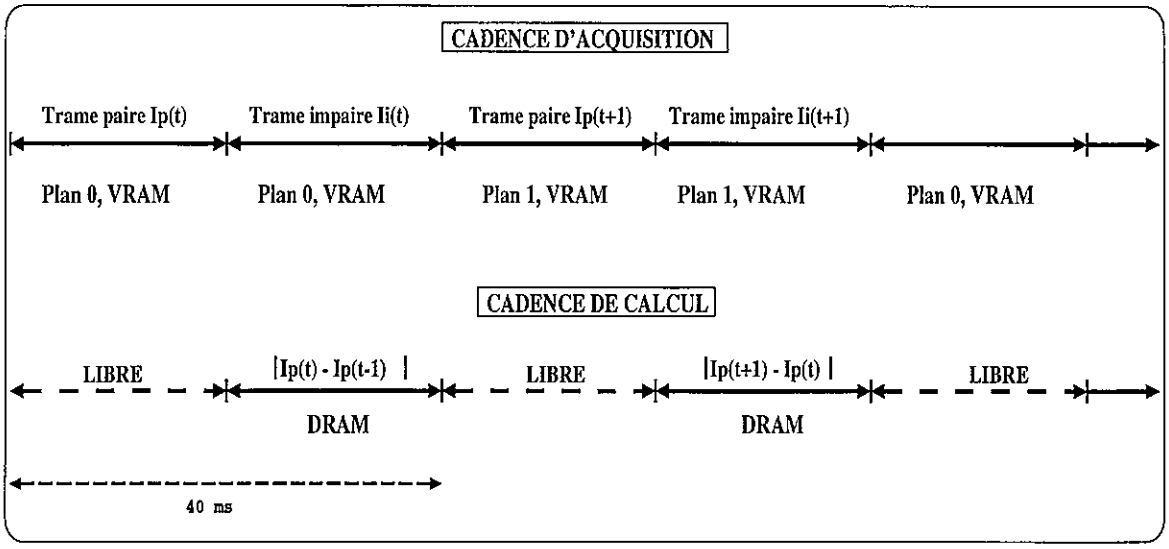

Figure 11. – Diagramme temporel pour l'acquisition et le prétraitement.

rapport à un seuil et non plus par utilisation d'un test de maximum de vraisemblance [10], l'objectif étant de limiter au maximum la charge de calcul. Evidemment, l'initialisation du champ des étiquettes en utilisant une méthode à base de tests de maximum de vraisemblance est de meilleure qualité. En effet, la prise de décision est faite relativement à un voisinage 3×3 alors que, dans le cas du seuillage de la différence d'images, la décision ne tient compte que de l'information au pixel considéré. Cette modification a une incidence sur la qualité des résultats lorsqu'il s'agit de détecter des objets mobiles faiblement texturés au déplacement lent par rapport à leur taille. Dans ce cas, l'initialisation est trop incomplète dans les zones de glissement des objets sur eux-même et les masques issus de la relaxation sont « troués » (la zone de glissement n'a pas été entièrement reconstruite). Le calcul des cartes binaires a lieu juste avant la relaxation.

4.2.2. Relaxation

Le calcul de l'étiquette en chacun des points s de l'image peut se décomposer en trois étapes :

- chargement des étiquettes utiles au calcul de l'énergie locale (8 voisins spatiaux, 2 voisins temporels) et de l'observation $o(s)$ correspondante;
- calcul de l'énergie pour l'étiquette mobile a et pour l'étiquette fixe b;
- choix de la nouvelle étiquette $e(s)$ correspondant à l'énergie locale la plus faible.

Le calcul des différentes énergies, ainsi que le choix de la nouvelle étiquette, ne font appel qu'à des tests de comparaison du type « if...then...else...endif »; ce genre de test n'étant pas aisément réalisable en assembleur, l'utilisation des différents espaces mémoire et des différentes possibilités d'adressage du processeur a permis d'optimiser au maximum l'implantation de ces tests.

Une remarque importante est à faire au sujet du caractère répétitif du calcul de l'énergie affectée à un point. En effet, une analyse précise des calculs d'énergies locales permet de constater que l'énergie du modèle u_m associée à l'étiquette fixe b est toujours l'opposé de celle associée à l'étiquette mobile a : $u_m(b) = -u_m(a)$. Il suffit donc de calculer $u_m(a)$ sans recalculer $u_m(b)$ d'où un organigramme de calcul simplifé (cf. figure 12). Le gain en temps de traitement est de $20t_i + 2t_l$ où t_l représente le temps de lecture/écriture mémoire et t_i le temps instruction.

4.2.3. évaluation des temps de calcul

Pour réaliser l'évaluation des temps de calcul, nous considérons l'organigramme de la figure 12. A chaque étape de cet organigramme, on affecte un temps théorique correspondant à l'exécution d'une opération (lecture/écriture, ALU) par cycle. De cette façon, nous disposons d'un modèle d'évaluation, indépendant de la carte utilisée pour la mise en oeuvre.

Le temps de calcul pour un point est donc : $t_p = 12t_l + 47t_i$.

Dans notre application, les valeurs numériques sont $t_l = 200ns$ (en mode page) et $t_i = 100ns$. Il en résulte pour chaque point un temps de calcul : $t_p = 7,1\mu s$. Les images traitées sont de taille 128×128 pixels ce qui conduit, en tenant compte des effets de bord (les premières lignes/colonnes n'étant pas traitées), à un temps $t_p * 126^2 = 110ms$ pour effectuer une itération du processus de relaxation sur toute l'image.

Comme nous l'avons vu lors de l'analyse théorique de la méthode de détection de mouvement, le nombre d'itérations nécessaire pour atteindre la convergence est peu élevé et une série de tests nous amène à considérer comme satisfaisant d'arrêter la relaxation après 4 itérations lors du traitement de scènes de trafic routier. Le

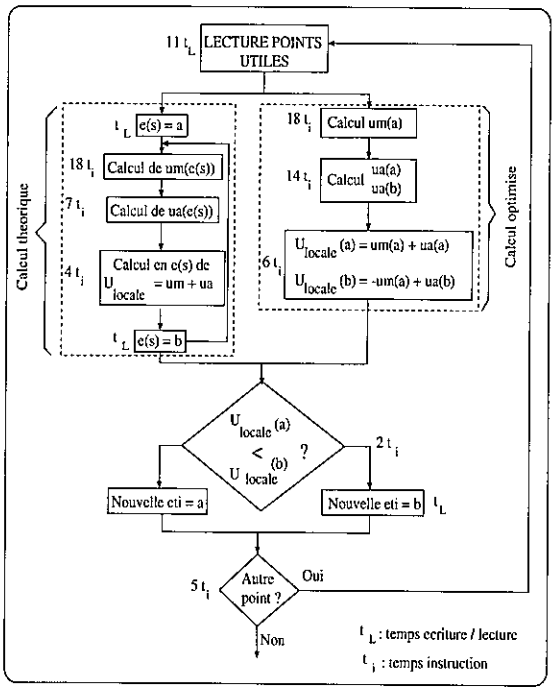

Figure 12. – Organigramme et temps de calcul.

temps de traitement global d'une image est alors de $4 * 110 = 440ms$. Le temps de calcul de la chaîne complète (en tenant compte du prétraitement) est donc de l'ordre de $T = 500ms$. Ce qui nous donne une **cadence théorique de 2 images/seconde**. Cette cadence peut sembler faible et éloignée de la cadence du traitement temps réel de 25 images par seconde. Soulignons cependant que l'implantation présentée a été réalisée sur une carte de traitement d'images du commerce [6]. La mise en oeuvre a dû respecter certaines contraintes imposées par la généralité de cette carte. Par exemple, la gestion vidéo est réalisée par le DSP lui-même ce qui occupe environ 15% du temps de calcul. Par ailleurs, le DSP n'est pas des plus performants (fréquence d'horloge 10MHz). Nous avons évalué que l'utilisation d'un DSP 96002 de Motorola (fréquence d'horloge 40MHz) conduirait à une cadence de traitement de 12 images par seconde pour des images de taille 256×256. Ceci fait l'objet de nos travaux actuels.

4.3. résultats

La mise en oeuvre sur la carte VPC941 de l'algorithme de détection de mouvement a permis d'atteindre en pratique une **cadence de 3 images/seconde**, c'est-à-dire plus que la prévision théorique et ceci grâce essentiellement aux diverses possibilités de parallélisation dues à l'architecture du DSP ST18941 (instructions à plusieurs champs, blocs mémoires internes indépendants, ...).

6. qui n'est pas particulièrement dédiée au problème de détection de mouvement.

Plusieurs types de séquences ont été traitées : des images de scènes naturelles, prises avec une caméra filmant une rue, ainsi que des images artificielles. Les figures 13 et 14 donnent une série de résultats sur des séquences d'images filmées dans la rue. Sur la figure 13, on constate que la relaxation a bien éliminé les observations isolées (bruit) et homogénéisé le masque de l'objet mobile. Les masques de la figure 14 sont relativement précis puisque l'algorithme détecte correctement le déplacement des jambes du piéton à chaque pas.

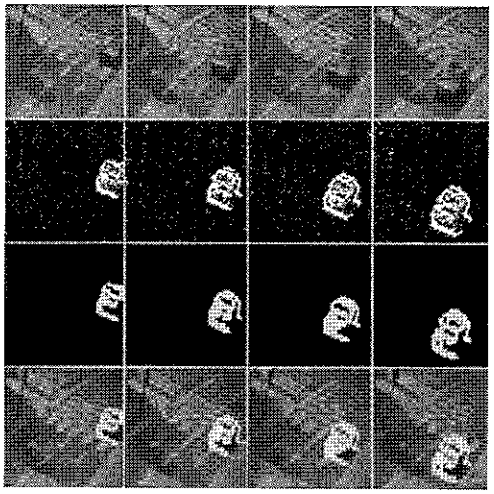

Figure 13. – Détection de mouvement : utilisation d'un DSP (les pixels mobiles sont représentés en gris et les fixes en noir). De haut en bas : 1) séquence d'images (une voiture en mouvement); 2) initialisations binaires; 3) masques des objets mobiles; 4) masques des objets mobiles superposés à la séquence d'images.

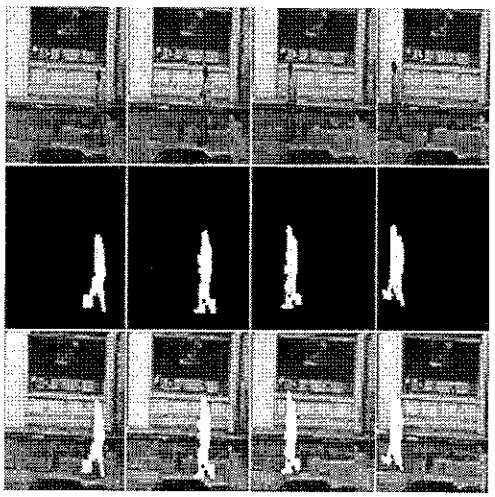

Figure 14. – Détection de mouvement : utilisation d'un DSP (les pixels mobiles sont représentés en gris et les fixes en noir). De haut en bas : 1) séquence d'images (un piéton en mouvement); 2) masques des objets mobiles; 3) masques des objets mobiles superposés à la séquence d'images.

Plusieurs remarques d'ordre général sont à faire quant aux résultats :

• d'une part, il faut noter l'importance du prétraitement, et plus particulièrement du seuillage visant à obtenir les cartes binaires servant à initialiser le champ d'étiquettes courant $\hat{E}_t$ et à estimer le champ futur $\hat{E}_{t+1}$. La qualité de cette initialisation permet évidemment une accélération de la convergence de l'algorithme, donc un gain non négligeable dans les temps de calculs. Cependant, nous n'avons pas trouvé de méthode de seuillage qui soit à la fois robuste et facile de mise en oeuvre sur le matériel utilisé.

• d'autre part, l'utilisation de RAM dynamique pour le stockage des différentes images engendre, de part sa nature, une perte de temps : les traitements se faisant par voisinage, chaque point traité nécessite deux lectures mémoire (une RAS, une CAS) et, de ce fait, les possibilités d'adressage indexé du processeur ne sont pas vraiment utilisées.

Les solutions envisageables sont :

• la mise en oeuvre d'un seuillage adaptatif mais à une cadence ad-hoc (par exemple, chaque seconde) correspondant aux réalités des scènes filmées (variations d'éclairement...);

• la séparation des tâches : prétraitement réalisé par des processeurs spécialisés (ASIC, FPGA), relaxation markovienne implantée sur un DSP performant utilisant des mémoires rapides (sans cycle d'attente).

5. conclusion

Le développement de notre algorithme de détection de mouvement selon une approche markovienne a été guidé par deux idées directrices :

• généraliser à moindre coût la détection binaire à un cadre de détection multi-étiquette, pour différencier les objets mobiles;

• fonctionner à une cadence rapide (si possible en temps réel).

Après avoir détaillé l'algorithme de détection de mouvement, nous avons montré que ce modèle pouvait être étendu au cas de la détection multi-étiquette. La relaxation gère automatiquement le choix entre plusieurs étiquettes mobiles possibles en fonction du voisinage considéré. L'algorithme ainsi défini ne permet cependant pas de traiter des cas difficiles tels que l'occultation d'objets ou la séparation de deux objets connexes. Les limitations rencontrées sont à mettre en relation avec la pauvreté de l'information de mouvement contenue dans les observations considérées (simple différence temporelle). L'algorithme décrit ici s'efforce d'utiliser au mieux cette information. Il ressort que pour obtenir un algorithme plus général de segmentation d'objets mobiles, il faut prendre en compte des informations de mouvement plus élaborées telles que le flux optique par exemple. Mais alors on se heurte à des difficultés de mise en oeuvre matérielle.

Nous nous sommes également intéressés à la mise en oeuvre de l'algorithme de détection sur une carte de traitement d'images à base de processeur de signal. La première réalisation effectuée sur une carte générique du commerce est encourageante (cadence de 3 images / seconde). Certes, la cadence de traitement obtenue est encore faible mais, en considérant une carte spécifique (séparation de la gestion vidéo, du prétraitement et de la relaxation markovienne) utilisant du matériel plus récent ou même simplement en utilisant un DSP plus performant (Motorola 96002 par exemple), une cadence de traitement de 12 images / seconde est tout à fait envisageable pour des séquences d'images de taille 256×256. Des recherches sont en cours à ce sujet.

BIBLIOGRAPHIE

[1] J. Besag "On the Statistical Analysis of Dirty Pictures". In Journal Royal Statistical Society, Vol. B-48, N.3, 1986, pp. 259-302.

[2] P. Bouthémy "Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images". In Techniques et Sciences informatiques, Vol.7, N.6, 1988, pp. 527-545.

[3] P. Bouthémy, P. Lalande "Recovery of moving object masks in an image sequence using local spatiotemporal contextual information". Optical Engineering, Vol.32, N.6, June 1993, pp.1205-1212.

[4] A. Caplier, F. Luthon "An MRF-based Spatiotemporal Multiresolution Algorithm for Motion Detection". Proc. of SCIA, Upsalla, Suède, Juin 1995, pp.158-162.

[5] A. Caplier, F. Luthon, C. Dumontier "Algorithme markovien de détection de mouvement. Mises en oeuvre "temps réel". Quinzième Colloque du GRETSI, Juan-les-Pins, France, Septembre 1995, pp.1033-1036.

[5bis] A. Caplier "Modèles markoviens de détection de mouvement dans les séquences d'images : approche spatio-temporelle et mises en oeuvre temps réel". Thèse de l'INP Grenoble, décembre 1995.

[6] P.B Chou, R. Raman "On Relaxation Algorithms Based on Markov Random Fields" In Tech. Rep. 212, Computer Science Department, Univ. of Rochester, July 1987.

[7] G.R. Cross, A.K. Jain "Markov Random Field Texture Models ". In Trans. Pattern Anal. and Machine Intel. Vol. PAMI-5, N.1, January 1983, pp. 25-39.

[8] 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.

[9] F. Heitz, P. Bouthémy " Multimodal Estimation of Discontinous Optical Flow Using Markov Random Fields ". In Trans. Pattern Anal. and Machine Intel., Vol. PAMI-15, N.12, December 1993, pp. 1217- 1232.

[10] Y.Z. Hsu, H.H. Nagel, G. Reckers "New Likelihood Test Methods for Change Detection in Image Sequences". In Computer Vision, Graphics and Image Processing, CVGIP-26, 1984, pp. 73-106.

[11] J. Hutchinson, C. Koch, C. Mead "Computing Motion Using Analog and Binary Resistive Networks". Computer, Vol.21, March 1988, pp. 52- 63.

[12] C. Koch, J. Marroquin, A. Yuille "Analog 'neuronal' networks in early vision". In Proc. Natl. Acad. Sci., USA, Biophysics, Vol.83, June 1886, pp. 4263-4267.

[13] P. Lalande "Détection du mouvement dans les séquences d'images selon une approche markovienne; application à la robotique sous-marine". Thèse de doctorat, Université de Rennes I, 1990.

[14] F. Luthon, A. Caplier " Motion Detection and Segmentation in image sequences using Markov Random Field Modelling ". In 4ème Eurographics Animation and Simulation Workshop, Barcelona, Spain, September 1993, pp. 265-275.

[15] 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.

[16] E. Mémin, F. Heitz "Algorithmes parallèles pour l'analyse d'images par champs markoviens". In Publication interne N.657, IRISA, Programme 4, Mai 1992, 74 pages.

[17] D.W. Murray, B.F. Buxton "Scene Segmentation from Visual Motion Using Global Optimization". In IEEE Trans. Pattern Anal. and Machine Intel. Vol. PAMI-9, N.2, January 1987, pp. 220-228.

[18] J. Zérubia, F. Ployette "Détection de contours et lissage d'images par deux algorithmes de relaxation. Mise en oeuvre sur la machine à connexions CM2". In Traitement du Signal, vol. 8, N.3, 3ème trimestre 1991, pp. 165-175.

Manuscrit reçu le 16 décembre 1994.

LES AUTEURS

Alice CAPLIER

Alice Caplier est diplômée de l'ENSIEG depuis 1991 et docteur de l'Institut National Polytechnique de Grenoble depuis 1995. Ses travaux de recherche au laboratoire de Traitement d'Images et de Reconnaissance des Formes (TIRF) concernent l'analyse du mouvement dans les séquences d'images par utilisation d'approches statistiques (champs de Markov) et/ou d'approches multi-résolution. Elle travaille également sur la mise en oeuvre temps réel d'un algorithme de détection de mouvement selon une approche markovienne.

Frank LUTHON

Franck Luthon est maître de conférences à l'ENSERG/INP Grenoble. Ses travaux de recherche au laboratoire de Traitement d'Images et de Reconnaissance des Formes portent sur l'analyse du mouvement et la compression de séquences d'images.

Christophe DUMONTIER

Christophe Dumontier est titulaire d'un DEA Signal Image Parole de l'Institut National Polytechnique de Grenoble. Il effectue actuellement une thèse dans le laboratoire de Traitement d'Images et de Reconnaissance des Formes. Ses travaux concernent l'étude et la mise en oeuvre temps réel d'un algorithme de détection de mouvement selon une approche markovienne. L'approche étudiée a abouti au développement d'une carte spécifique, d'architecture pipe-line, utilisant des composants de logique programmables (FPGA) et un processeur de signal (DSP).

Pierre-Yves COULON

Pierre-Yves Coulon est maître de conférences à l'ENSERG/INP Grenoble. Ses thèmes de recherche se situent dans les domaines du suivi du mouvement et des architectures matérielles. Actuellement, ses travaux au laboratoire de Traitement d'Images et de Reconnaissance des Formes portent sur l'architecture de machine pour la vision et sur les problèmes de coopération forme-mouvement en traitement d'image.