Détection de mouvement dans les séquences d'images. Mises en oeuvre temps réel.

Franck Luthon

▶ **To cite this version:**

Franck Luthon. Détection de mouvement dans les séquences d'images. Mises en oeuvre temps réel.. A. Chéhikian, P.Y. Coulon, Franck Luthon. 2nde session de l'Ecole des Techniques Avancées en Signal-Image-Parole (ETASIP'97), pp.181-201, 1997, ETASIP 97. ⟨hal-02330248⟩

HAL Id: hal-02330248

https://hal.science/hal-02330248v1

Submitted on 23 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

Détection de mouvement dans les séquences d'images. Mises en œuvre temps réel.

Franck Luthon
Laboratoire de Traitement d'Images et Reconnaissance de Formes
INPG, 46 av. Félix-Viallet, 38031 Grenoble Cedex, France
email : luthon@tirf.inpg.fr fax : +33 (0)4 76 57 47 90

1 Introduction

L'analyse du mouvement dans les séquences d'images est un domaine de recherche active à l'heure actuelle, en raison de son importance dans de nombreuses applications : télésurveillance, compression pour les télécommunications ou l'archivage, diagnostic médical, météorologie, contôle non destructif, robotique mobile, etc. On distingue habituellement quatre phases en analyse de mouvement : la *détection* (des zones mobiles), l'*estimation* (des vecteurs-vitesses, en chaque pixel ou pour chaque objet), la *segmentation* (en zones cohérentes au sens du mouvement) et l'*interprétation* de haut niveau (reconnaissance de formes faisant appel à l'intelligence artificielle). Ces quatre étapes ne sont en aucun cas indépendantes ni forcément séquentielles, mais au contraire fortement interdépendantes (notamment en ce qui concerne les deux phases estimation-segmentation). Nous nous intéressons ici uniquement à la phase de détection du mouvement des objets mobiles dans le cas d'une caméra fixe par rapport à la scène observée.

2 Principes

La détection de mouvement consiste à attribuer à chaque pixel ou *site* $s = (x, y, t)$ des images d'une séquence un attribut, qu'on appelle *étiquette* e_s, indiquant si le pixel appartient à un objet mobile ou au fond fixe de la scène observée :

$$e_s = e(x, y, t) = \begin{cases} a = \text{"1"} & \text{si le pixel appartient à un objet mobile,} \\ b = \text{"0"} & \text{si le pixel appartient au fond fixe.} \end{cases} \quad (1)$$

En faisant l'hypothèse d'une caméra fixe et d'un éclairement quasi-constant de la scène observée, on peut extraire en chaque pixel une information de bas niveau, qu'on appelle *obersvation* o_s, portant sur la variation temporelle de l'intensité lumineuse I du pixel :

$$o_s = o(x, y, t) = |I(x, y, t) - I(x, y, t-1)| \quad (2)$$

Dans le cas idéal, cette variation temporelle entre deux instants d'acquisition $t-1$ et t est nulle pour un pixel du fond fixe, non nulle s'il y a eu mouvement d'un objet d'intensité non parfaitement uniforme. Cependant, cette observation de bas niveau est peu robuste (bruit d'acquisition, etc.) et ne donne qu'une information partielle sur les objets en mouvement. Un simple seuillage ne permet pas d'extraire les objets mobiles. En effet, on distingue typiquement 4 zones dans le champ d'observations (Fig. 1) :

- le fond fixe,
- la zone d'écho (zone de fond découverte par l'objet à l'instant t),
- la zone de glissement de l'objet sur lui-même,
- la zone de recouvrement (zone de fond recouverte par l'objet à l'instant t).

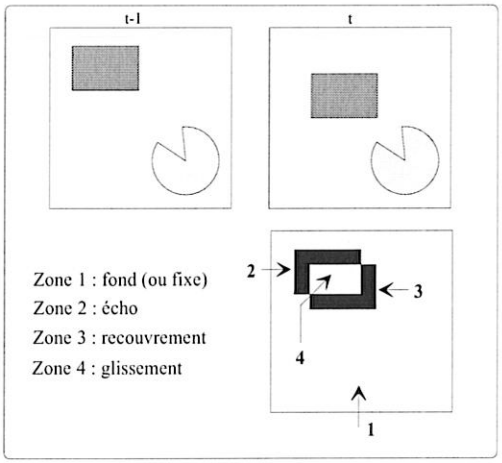

Figure 1: Carte binaire des changements temporels entre les instants $t-1$ et t. Cas d'un rectangle mobile et d'un camembert fixe.

Il faut d'une part éliminer l'écho, d'autre part reconstruire l'intérieur des objets mobiles (zone de glissement). Différentes techniques, très simples dans leur principe, mais relativement peu robustes, permettent d'obtenir les zones mobiles. Citons entre autres :

- la technique de simple différence par rapport à une image de référence censée ne contenir que le fond fixe de la scène : le problème est bien sûr l'obtention de cette image de référence, qu'il faut éventuellement mettre à jour régulièrement en fonction de l'évolution de la scène,
- la technique du ET logique entre cartes binaires de changements temporels, qui ne donne de bons résultats que si le mouvement est assez rapide pour qu'il n'y ait pas de recouvrement entre deux positions successives de l'objet, ce qui est assez restrictif.

C'est pourquoi on fait appel à une approche statistique probabiliste pour régulariser le problème qui est initialement mal posé (sous-déterminé). Le principe consiste à introduire une modélisation a priori du champ des étiquettes, à l'aide de la théorie des champs aléatoires de Markov (*Markov Random Fiels* ou MRF en anglais) [6]. On contraint alors la solution vers une configuration la plus probable du champ des étiquettes étant donné les observations et le modèle a priori [3].

3 Approche markovienne

3.1 Fonctions d'énergie

Adoptons les notations suivantes :

- S l'image à l'instant courant t,
- s un pixel (*site*) quelconque de S,
- η_s un voisinage spatio-temporel de s, par exemple celui défini par la Fig. 2,
- r n'importe quel voisin de s (spatial ou temporel),
- C l'ensemble des cliques binaires $c = (s, r)$ constituant les voisinages η_s,
- $O = \{O_s, s \in S\}$ le champ aléatoire des observations,
- $E = \{E_s, s \in S\}$ le champ aléatoire des étiquettes,
- $o = \{o_s, s \in S\}$ une réalisation particulière du champ d'observations O à l'instant t,
- $e = \{e_s, s \in S\}$ une réalisation particulière du champ d'étiquettes E à l'instant t,
- R l'ensemble des configurations possibles e du champ aléatoire E.

A chaque fois que ce sera nécessaire, on ajoutera un indice temporel pour préciser l'instant considéré : t, $t-1$, etc.

Une clique binaire est une paire de sites mutuellement voisins. Les interactions spatiales et temporelles entre étiquettes sont modélisées par un MRF, qui constitue le modèle a priori du champ des étiquettes. La propriété essentielle d'un MRF relativement à un voisinage est le caractère local des interactions : la probabilité d'avoir en un pixel s une étiquette e_s ne dépend que des étiquettes de ses voisins, et non pas de toute l'image.

1. $\forall e \in R, Pr[E = e] > 0$
2. $Pr\left[E_s = e_s \;/\; E_r = e_r, r \neq s, r \in S\right] = Pr\left[E_s = e_s \;/\; E_r = e_r, r \in \eta_s\right]$

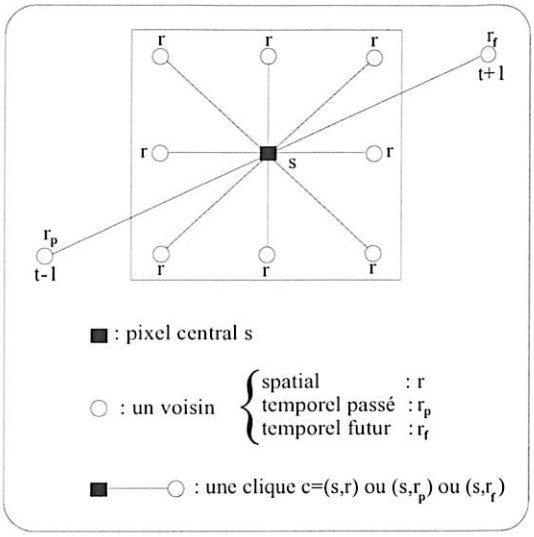

Figure 2: Voisinage spatio-temporel et cliques binaires associées.

Cette propriéte est cruciale car elle implique des calculs purement locaux et fortement parallélisables, d'où les possibilités de mise en œuvre temps réel [4], [8].

Un MRF étant équivalent à une distribution de Gibbs, on peut exprimer la probabilité a priori du champ d'étiquettes à l'aide d'une fonction d'énergie :

$$Pr[E = e] = \frac{1}{Z}\, exp(-U_m(e)) \tag{3}$$

où Z est une constante de normalisation (appelée *fonction de partition*) et U_m une fonction d'énergie associée au modèle a priori. Le champ d'étiquettes le plus probable relativement aux observations est obtenu par le critère du Maximum A Posteriori (MAP). A l'aide du théorème de Bayes et de l'équation (3), on montre que ce critère équivaut à la minimisation d'une fonction d'énergie totale U [9] :

$$\max_e Pr[E = e/O = o] \Longleftrightarrow \min_e U \quad \text{où} \quad U = U_m(e) + U_a(o, e) \tag{4}$$

$U_m(e)$ est un terme d'énergie qui assure la régularisation de la solution. Il s'exprime comme une somme de potentiels énergétiques élémentaires sur les cliques :

$$U_m(e) = \sum_{c \in C} V_c(e_s, e_r) \tag{5}$$

Ces potentiels élémentaires sont définis en fonction du problème à traiter. On peut par exemple prendre des potentiels à niveaux du type :

$$V_c(e_s, e_r) = \begin{cases} -\beta & \text{si } e_s = e_r \\ +\beta & \text{si } e_s \neq e_r \end{cases} \tag{6}$$

ou des potentiels quadratiques du type :

$$V_c(e_s, e_r) = \beta(e_s - e_r)^2 \tag{7}$$

où $\beta > 0$ prend une des trois valeurs β_s, β_p ou β_f selon la clique considérée (spatiale, passée ou future). Ces potentiels énergétiques favorisent un étiquetage homogène puisque des étiquettes semblables sur des pixels voisins induisent une moindre contribution énergétique, donc une configuration plus favorable. En pratique, on prend $\beta_f > \beta_p$, pour bien éliminer les zones d'écho du mouvement.

$U_a(o, e)$ est l'énergie d'adéquation, qui assure une bonne attache aux données. Elle traduit le lien entre observations et étiquettes et s'exprime à l'aide d'une fonction Ψ censée modéliser les observations :

$$U_a(o, e) = \frac{1}{2\sigma^2} \sum_{s \in S} [o_s - \Psi(e_s)]^2 \tag{8}$$

où σ^2 est la variance des observations et Ψ est une fonction d'attache aux observations définie en fonction du problème à traiter. On peut par exemple prendre une fonction simple du type :

$$\Psi(e_s) = \begin{cases} 0 & \text{si } e_s = b \\ \alpha > 0 & \text{sinon} \end{cases} \tag{9}$$

où α correspond alors à la valeur moyenne des observations non nulles.

3.2 Estimation des paramètres

Tous les paramètres qui interviennent dans la modélisation (α, β, σ) peuvent soit être déterminés de façon empirique après des tests sur des séquences typiques correspondant à l'application envisagée, soit être estimés automatiquement grâce à des algorithmes du type EM (*Expectation-Maximisation*) ou SEM (*Stochastic Expectation Maximisation*). Voir par exemple les travaux de Pieczynski [1] [13].

3.3 Algorithmes de relaxation

Pour calculer la configuration d'étiquettes donnant l'énergie minimale, différents algorithmes, dits de relaxation, peuvent être utilisés.

- Les algorithmes de relaxation stochastiques, du type recuit simulé, consistent à explorer l'espace des configurations possibles avec une loi de descente en température adéquate (i.e. suffisamment lente) qui assure en principe de converger vers le minimum global de la fonction d'énergie, mais au prix d'un coût de calcul prohibitif.

- C'est pourquoi on leur préfère souvent les algorithmes de relaxation déterministes, du type ICM (*Iterated Conditional Modes*) [2], beaucoup moins lourds en calculs mais qui ne garantissent pas de converger vers le mimimum global d'énergie. Ils risquent de rester piégés dans un minimum local, car ils n'autorisent aucune remontée en température pour sortir d'un minimum local. C'est pourquoi ce type d'algorithme requiert une bonne configuration initiale du champ d'étiquettes (Fig. 3).

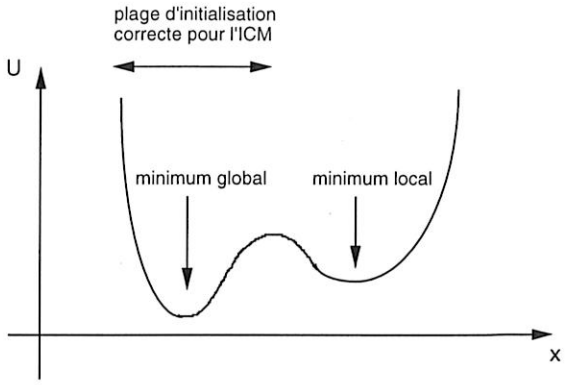

Figure 3: Influence de l'initialisation sur le résultat de l'ICM : évolution de la fonction d'énergie U en fonction de la configuration X du champ d'étiquettes.

Ces algorithmes déterministes sont récursifs et itératifs : on visite chaque pixel de l'image et on calcule, pour chaque étiquette qu'on peut lui attribuer, la contribution énergétique sur son voisinage. On retient l'étiquette qui donne l'énergie minimale, puis on passe au pixel suivant et ainsi de suite sur toute l'image. On réitère la visite des sites de l'image jusqu'à la convergence de la fonction d'énergie totale U vers une valeur qui n'évolue plus, ou très peu, au cours des itérations. Différentes politiques de visite de sites sont envisageables :

- visite séquentielle classique (*line scanning*) (ligne par ligne de gauche à droite et image par image de haut en bas),
- visite aléatoire,
- visite chaînée (ligne par ligne alternativement de gauche à droite puis de droite à gauche et image par image alternativement de haut en bas puis de bas en haut) pour bénéficier au mieux des derniers voisinages mis à jour.

Différentes politiques de mise à jour des sites sont aussi envisageables, le choix dépendant notamment du type de matériel utilisé pour la mise en œuvre :

- pixel-récursif : on met à jour chaque pixel au fur et à mesure de la visite,
- ligne-récursif : on attend d'avoir visité tous les pixels d'une ligne avant de mettre à jour leurs étiquettes,
- image-récursif : on attend d'avoir visité toute une image avant de faire une mise à jour simultanée de toutes les étiquettes de l'image.

De même, différents critères d'arrêt des itérations peuvent être utilisés :

- nombre de pixels instables inférieur à un seuil prédéterminé,
- variation relative d'énergie totale inférieure à un seuil prédéterminé,
- nombre fixe d'itérations (en pratique un faible nombre d'itérations, typiquement 5 à 10 itérations par image, est suffisant). Ce critère est évidemment le moins coûteux pour une mise en œuvre matérielle.

3.4 Synoptique de l'algorithme de détection de mouvement

Le synoptique complet de l'algorithme est donné Fig. 4. Sur ce schéma-bloc, la notation $\hat{E}$ représente un champ d'étiquettes initial grossièrement estimé par simple binarisation du champ correspondant d'observations. Pour réaliser la binarisation, on peut soit faire un simple seuillage, soit utiliser des techniques plus robustes de maximum de vraisemblance, avec modèle de luminance (constant, linéaire ou quadratique) sur un voisiange des pixels [10]. Etant donné le voisinage spatio-

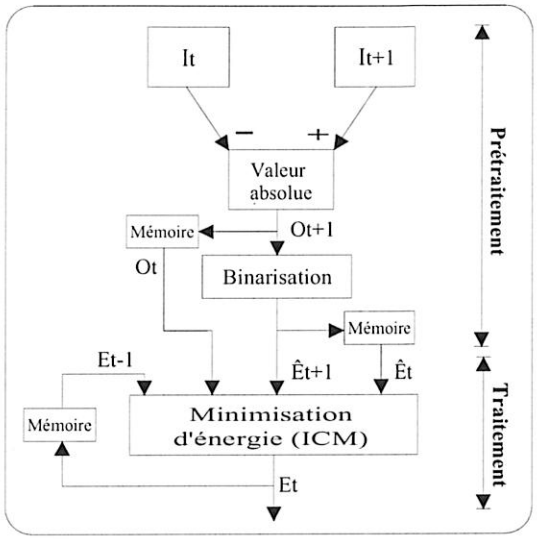

Figure 4: Synoptique de l'algorithme de détection de mouvement.

temporel utilisé pour détecter le mouvement, voisinage qui inclut un voisin futur et un voisin passé pour chaque pixel (Fig. 2), le résultat de traitement est obtenu avec un retard d'une image par rapport à l'acquisition, puisqu'il faut disposer de l'image en $t+1$ pour estimer le champ d'étiquettes en t.

4 Multirésolution spatio-temporelle

Pour améliorer les performances de l'algorithme dans le cas du mouvement d'objets très lents (mouvement sub-pixel), ou de gros objets peu texturés, on peut envisager d'utiliser une technique du type multirésolution. On présente ici quelques résultats obtenus avec une technique originale de multirésolution spatio-temporelle qui consiste à filtrer passe-bas et sous-échantillonner la séquence d'images à la fois en espace et en temps selon le schéma de la Fig. 5, où est aussi représenté le noyau du filtre 3D appliqué à la séquence.

Un exemple de pyramide spatio-temporelle est donné Fig. 6. Le nombre de niveaux dans la pyramide est fonction notamment de l'amplitude des mouvements présents dans la scène.

Le filtrage spatial permet de renforcer les observations dans le cas de gros objets uniformes (Fig. 7).

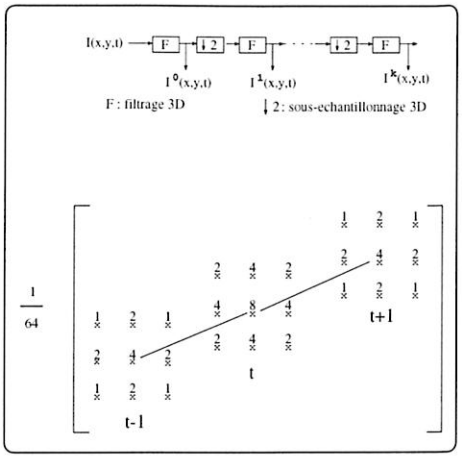

Figure 5: Principe de construction de la pyramide et noyau du filtre spatio-temporel 3D.

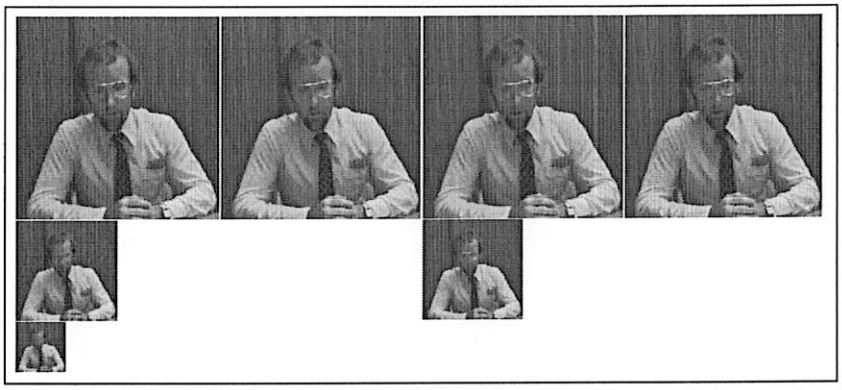

Figure 6: Exemple de pyramide spatio-temporelle à trois niveaux sur la séquence *Trevor* (de haut en bas, $k = 0, 1, 2$).

Quant au filtrage temporel, il permet d'intégrer l'information de mouvement de plusieurs images successives, ce qui est indispensable pour détecter des objets au mouvement très lent (Fig. 8).

5 Voisinage 3D spatio-temporel

On peut envisager une structure de voisinage 3D comme présenté Fig. 9 [5]. Si on note $\delta_x, \delta_y, \delta_t$ les coordonnées du vecteur représentatif de chaque clique dans l'espace 3D (x, y, t) centré en le pixel courant s, on définit :

- 4 cliques spatiales horizontales et verticales (δ_x ou $\delta_y = \pm 1$, $\delta_t = 0$);
- 4 cliques spatiales diagonales (δ_x et $\delta_y = \pm 1$, $\delta_t = 0$);
- 2 cliques temporelles (δ_x et $\delta_y = 0$, $\delta_t = \pm 1$);
- 8 cliques spatio-temporelles horizontales et verticales (δ_x ou $\delta_y = \pm 1$, $\delta_t = \pm 1$);

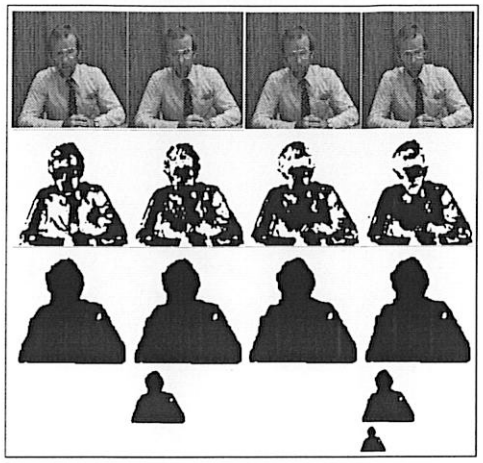

Figure 7: Détection de mouvement multirésolution : de haut en bas : 1) séquence *Trevor*; 2) masques monorésolution; 3) masques multirésolution (3 niveaux d'analyse $k = 0, 1, 2$).

- 8 cliques spatio-temporelles diagonales (δ_x et $\delta_y = \pm 1$, $\delta_t = \pm 1$).

Il faut alors modéliser les interactions spatio-temporelles sur toutes les cliques de ce voisinage 3D par des potentiels $\beta(s,r)$ qui sont fonction du type de clique $c = (s,r)$ considérée :

$$\beta(s,r) = \frac{1}{d^2(s,r)\,(\frac{\delta_x(s,r)^2}{\beta_s} + \frac{\delta_y(s,r)^2}{\beta_s} + \frac{\delta_t(s,r)^2}{\beta_t})} \tag{10}$$

où $d(s,r) = \sqrt{\delta_x^2 + \delta_y^2 + \delta_t^2}$ est la distance euclidienne entre le pixel courant s et le voisin considéré r. Cette formule donne :

- $\beta(s,r) = \beta_s$ pour les cliques spatiales horizontales ou verticaes ($d(s,r) = 1$);
- $\beta(s,r) = \frac{\beta_s}{4}$ pour les cliques spatiales diagonales ($d(s,r) = \sqrt{2}$);
- $\beta(s,r) = \beta_t$ pour les cliques temporelles ($d(s,r) = 1$);
- $\beta(s,r) = \frac{\beta_s \beta_t}{2(\beta_s + \beta_t)}$ pour les cliques spatio-temporelles horizontales ou verticales ($d(s,r) = \sqrt{2}$);
- $\beta(s,r) = \frac{\beta_s \beta_t}{3(\beta_s + 2\beta_t)}$ pour les cliques spatio-temporelles diagonales ($d(s,r) = \sqrt{3}$).

Notons que cette définition de $\beta(s,r)$ ne fait intervenir que deux paramètres β_s et β_t.

Cette variante de l'algorithme permet d'améliorer les performances dans le cas de séquences bruitées comme illustré sur la Fig. 10.

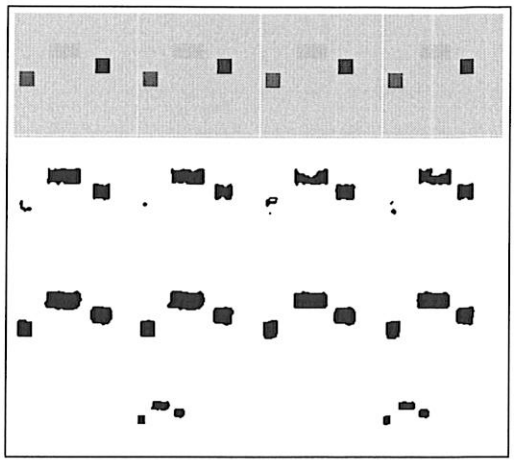

Figure 8: Détection de mouvement sous-pixel : de haut en bas : 1) séquence synthétique avec 3 objets mobiles dont l'un a un mouvement sub-pixel; 2) masques monorésolution; 3) masques multirésolution (2 niveaux d'analyse $k = 0, 1$).

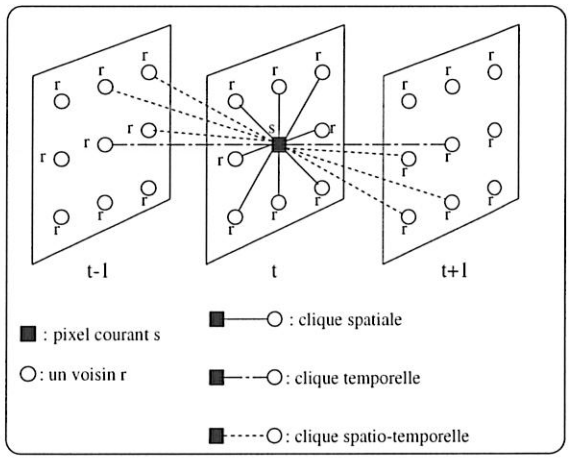

Figure 9: Voisinage spatio-temporel 3D, cliques binaires associées (pour des raisons de clarté, toutes les cliques possibles n'ont pas été représentées).

6 Mises en œuvre matérielles

L'algorithme de détection de mouvement se décompose en deux étapes principales (cf. Fig. 4):

- un prétraitement qui calcule les observations et les champs initiaux d'étiquettes,
- le traitement proprement dit qui calcule le minimum d'énergie sur les voisinages spatio-temporels.

Le prétraitement qui consiste simplement en des calculs de différences d'images, de valeurs absolues et des binarisations par seuillage ne pose pas de problème particulier d'implantation temps réel. Par contre, la minimisation d'énergie est un processus itératif gourmand en calculs et en mémoire et doit donc faire l'objet d'une mise en

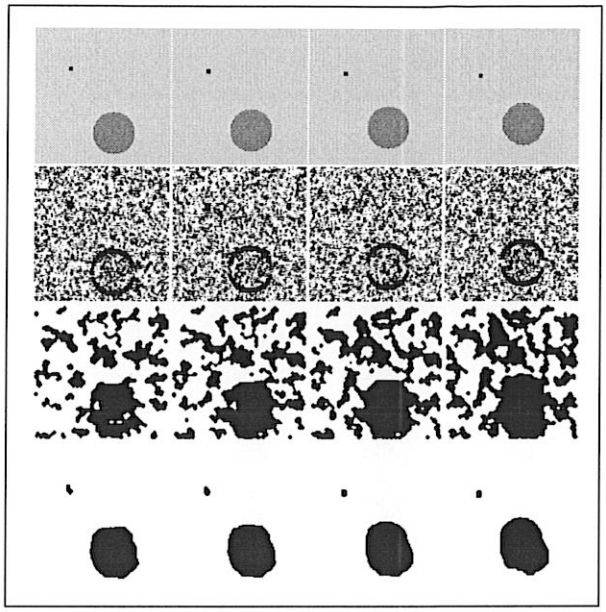

Figure 10: Comparaison des deux algorithmes : de haut en bas : 1) séquence synthétique bruitée; 2) initialisation binaire; 3) masques avec l'algorithme spatial; 4) masques avec l'algorithme spatio-temporel.

œuvre sur une architecture adaptée. Une évaluation grossière montre en effet que la relaxation markovienne compte pour 90% de la charge totale de calcul. On a alors affaire à un problème d'adéquation algorithme-achitecture. Plusieurs mises en œuvre sont envisageables :

- la première solution, qui prend essentiellement en compte le caractère parallèle des calculs identiques effectués en chaque pixel, consiste à implanter l'algorithme sur une machine parallèle SIMD ou MIMD. L'inconvénient principal de ce type de mise en œuvre est l'encombrement, le coût et les transferts de données.

- une seconde solution, qui prend essentiellement en compte le caractère local des calculs sur un petit voisinage spatio-temporel, consiste à s'inspirer du fonctionnement des réseaux de neurones pour implanter l'algorithme sur un circuit résistif analogique en technologie VLSI. Il faut alors développer une analogie électrique du modèle markovien. Inconvénients : coût de développement. Avantages : taille réduite du circuit.

- une troisième solution originale, développée au sein du laboratoire TIRF, consiste à implanter l'algorithme sur une carte au format PC à base de DSP et de circuits logiques programmables FPGA. Avantages : coût réduit, encombrement réduit. Inconvénient : manque de flexibilité.

6.1 Machines parallèles

La mise en œuvre des algorithmes markoviens sur une machine de type parallèle est une solution assez naturelle pour exploiter au mieux le parallèlisme intrinséque à ce type de traitement.

Un exemple de machine parallèle développée pour la détection de mouvement est la machine Transvision [7]. Nous donnons ici le synoptique général de cette machine (Fig. 11). L'architecture de la machine Tranvision est centrée sur l'utilisation

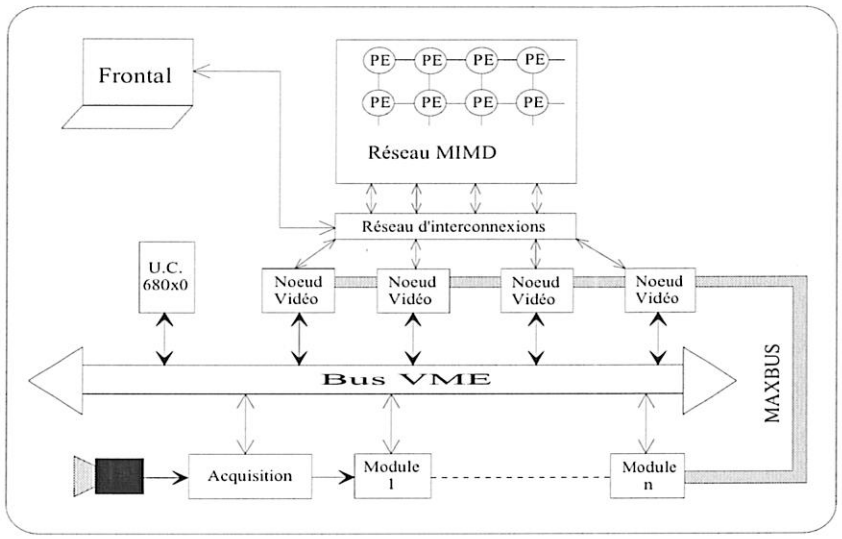

Figure 11: Synoptique général de la machine Transvision.

de processeurs spécifiques, les transputers. Basée sur l'utilisation conjointe d'un monde pipeline et d'un monde MIMD, la machine Transvision met en œuvre un concept de noeud vidéo faisant office de passerelle entre ces deux mondes. Chaque noeud vidéo comporte un transputer et une mémoire vidéo double port accessible par les deux mondes pipeline et MIMD. Le mécanisme de synchronisation est basé sur le concept de *rendez-vous* OCCAM et permet d'alimenter et de resynchroniser le réseau MIMD sans goulet d'étranglement.
Les modules pipeline et MIMD sont gérés via un bus VME et une unité centrale Motorola 68040 fonctionnant sous VxWork. Le développement d'applications sur les transputers est réalisé sur station de travail ou sur PC en langage OCCAM ou C parallèle.
Le module pipeline considéré dans cette application se restreint à une carte d'acquisition vidéo de la société DataCube permettant l'acquisition et la digitalisation d'images 512 x 512 points en temps réel. Ces images sont alors transmises aux noeuds vidéo par l'intermédiaire d'un réseau MaxBus.
Le module MIMD est constitué de douze transputers Inmos T800.
Cette machine fournit donc une chaîne complète de traitement, de l'acquisition vidéo à l'interprétation des résultats.

Indépendamment des performances très intéressantes de cette machine, cette description sommaire met en avant la complexité architecturale d'un tel système, et

donc a fortiori son encombrement et son coût. C'est pourquoi il peut être intéressant d'envisager des solutions alternatives comme celles décrites ci-dessous.

6.2 Circuit VLSI analogique

Un processus de minimisation d'énergie peut être efficacement implanté matériellement sur un réseau résistif analogique qu'on laisse relaxer jusqu'à son état d'équilibre (dissipation minimale d'énergie). Mais une telle approche nécessite d'adapter le modèle initial pour trouver une bonne analogie électrique avec le comportement d'un réseau résistif.

Nous avons défini un modèle qui respecte cette contrainte [12]. Les pixels sont matérialisés par les noeuds interconnectés d'un réseau électrique résistif. On remplace le champ d'étiquettes binaires (a, b) par un champ d'étiquettes continues variant de "0" (étiquette b) à "1" (étiquette a). Ces étiquettes continues sont alors représentées par des potentiels électriques variant de la tension de masse à la tension d'alimentation du circuit analogique. En utilisant un voisinage d'ordre 1 (4 voisins spatiaux et 2 voisins temporels), des potentiels énergétiques quadratiques (cf. Eq.(7)), et une fonction Ψ judicieuse [12], on montre qu'on aboutit à une expression de l'énergie sous la forme :

$$\begin{aligned} U \quad = \sum_{i,j} \quad & \beta_s(e_{ij} - e_{i+1,j})^2 + \beta_s(e_{ij} - e_{i-1,j})^2 + \beta_s(e_{ij} - e_{i,j+1})^2 + \beta_s(e_{ij} - e_{i,j-1})^2 + \\ & \beta_p(e_{ij} - p_{ij})^2 + \beta_f(e_{ij} - f_{ij})^2 + K\left[o_{ij} - \Psi(e_{ij}, p_{ij})\right]^2 \end{aligned} \tag{11}$$

où p_{ij} et f_{ij} sont les potentiels électriques des voisins passé et futur, e_{ij} celui du pixel (i,j) et K une constante.

Le minimum d'énergie correspond à l'annulation des dérivées partielles par rapport à tous les potentiels électriques e_{ij} en chaque noeud (i,j) du réseau. On obtient alors une équation dont l'analogie électrique est immédiate :

$$\forall ij \quad C\frac{\partial e_{ij}}{\partial t} = \beta_s \nabla^2 e_{ij} + \beta_{pk}(e_{ij} - p_{ij}) + \beta_f(e_{ij} - f_{ij}) + K\alpha(p_{ij} - e_0)o_{ij} \tag{12}$$

où $\nabla^2 e_{ij} = 4e_{ij} - e_{i+1,j} - e_{i-1,j} - e_{i,j+1} - e_{i,j-1}$ est une approximation discrète du Laplacien et β_{pk} et $K\alpha$ sont des constantes. La cellule élémentaire associée à chaque site est donnée Fig. 12. L'idée principale du fonctionnement de la cellule électrique consiste à injecter ou pomper du courant en chaque noeud du réseau (grâce à un générateur de courant commandé), pour faire monter ou baisser le potentiel électrique du noeud, selon que l'amplitude de l'observation correspondante est forte (pixel mobile) ou faible (pixel fixe).

On peut intégrer sur chaque cellule élémentaire du réseau un photorécepteur, ce qui fait du circuit électrique une caméra "intelligente". L'architecture du réseau analogique résultant est analogue à celle d'une caméra CCD et est donnée Fig. 13.

Un exemple typique de résultat est présenté Fig. 14.

L'avantage d'une telle mise en œuvre matérielle est l'encombrement réduit du système et sa vitesse de traitement (relaxation du réseau en quelques $10ns$). Les inconvénients sont le coût élévé d'un prototype et les problèmes technologiques d'intégration VLSI à résoudre (taille de la cellule, consommation, interconnexions, etc.).

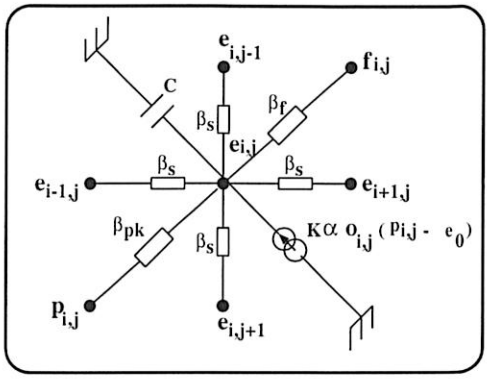

Figure 12: Cellule analogique élémentaire.

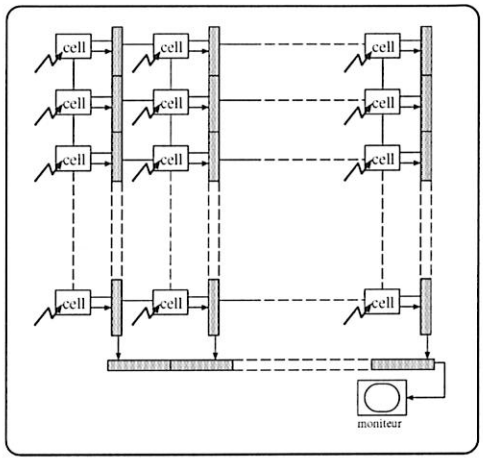

Figure 13: Architecture du réseau VLSI analogique

6.3 Cartes à base de DSP et FPGA

Après une étude détaillée de l'algorithme de détection de mouvement, il s'avère possible de proposer une architecture matérielle alternative aux solutions coûteuses présentées plus haut. On peut en effet réaliser le prétraitement avec des circuits logiques programmables (FPGA) et, grâce aux processeurs de signaux numériques actuellement disponibles sur le marché, mettre en œuvre le traitement proprement dit (la relaxation énergétique) à moindre frais. L'architecture de la carte développée est donnée Fig.15. Tout ce qui concerne l'acquisition, le prétraitement et la visualisation fonctionne à la cadence pixel (15 MHz). Seule la relaxation markovienne est réalisée en mode asynchrone par un DSP (Motorola 96002), avec une mémoire tampon qui sert d'interface. Une carte opérationnelle, qui traite des images de taille 256x256 à la cadence de 15 images par seconde, a été développée au laboratoire TIRF (Fig. 16). Cette carte s'insère sur le bus ISA d'un ordinateur personnel, est fonctionne de manière complètement autonome après chargement de son programme (phase initiale de *boot*). On dispose ainsi d'un prototype complet et peu coûteux, allant de l'acquisition à la visualisation des masques des objets mobiles, et approchant un fonctionnement en temps réel (Fig. 17).

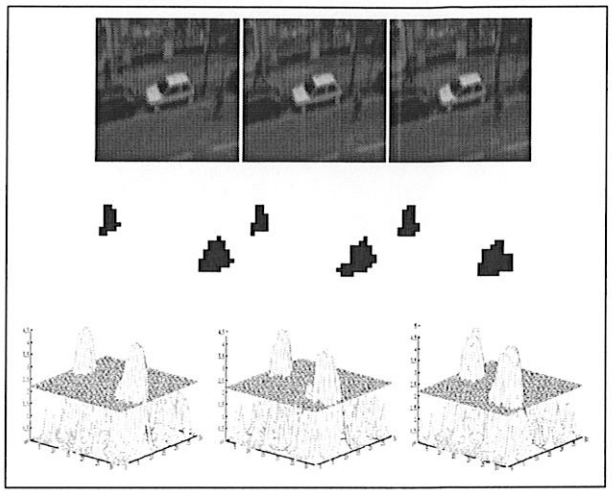

Figure 14: Simulation électrique. De haut en bas : 1) séquence naturelle avec un vélo et un piéton mobiles ; 2) masques binaires détectés ; 3) potentiels électriques.

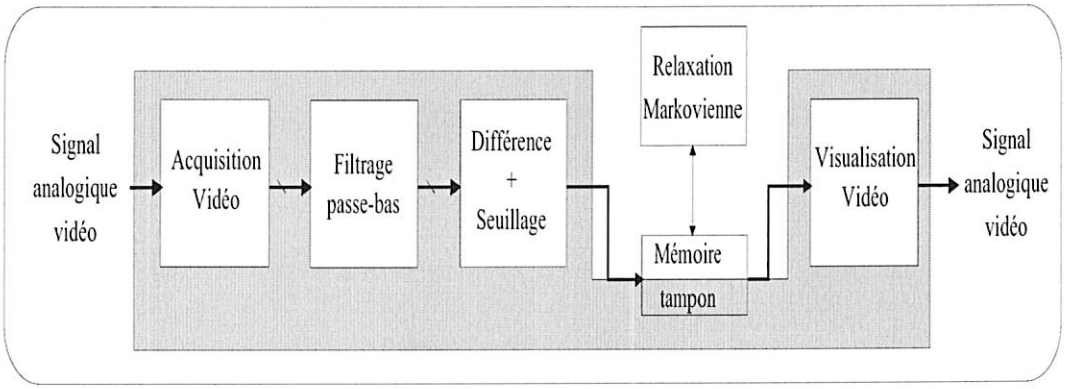

Figure 15: Architecture matérielle proposée. La partie grisée correspond aux étages fonctionnant en mode synchrone c'est-à-dire à la cadence du flux pixel.

7 Exemples d'applications

7.1 Télésurveillance

Une application typique de détection de mouvement est le contrôle du trafic routier ou la télésurveillance à l'entrée d'un site grâce à une caméra "intelligente". La Fig. 18 présente un exemple de résultat de détection automatique du mouvement d'un piéton sur un trottoir, obtenu grâce à la carte à DSP décrite ci-dessus. Notons que l'algorithme détecte également l'ombre du piéton sur le capot de la voiture en stationnement (il ne s'agit pas d'une fausse détection).

7.2 Analyse du mouvement des lèvres d'un locuteur

Il est bien connu que l'homme s'aide d'informations visuelles pour améliorer sa reconnaissance de la parole, notamment dans un environnement bruité. En complément

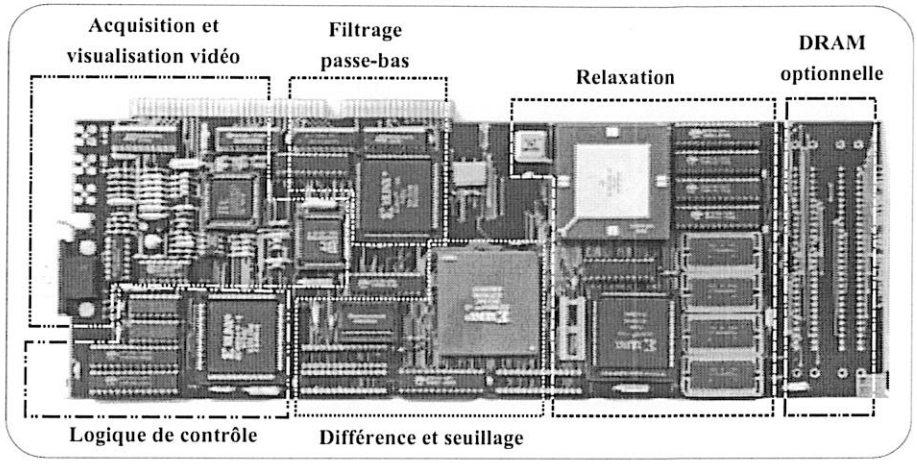

Figure 16: Photo de la carte prototype développée.

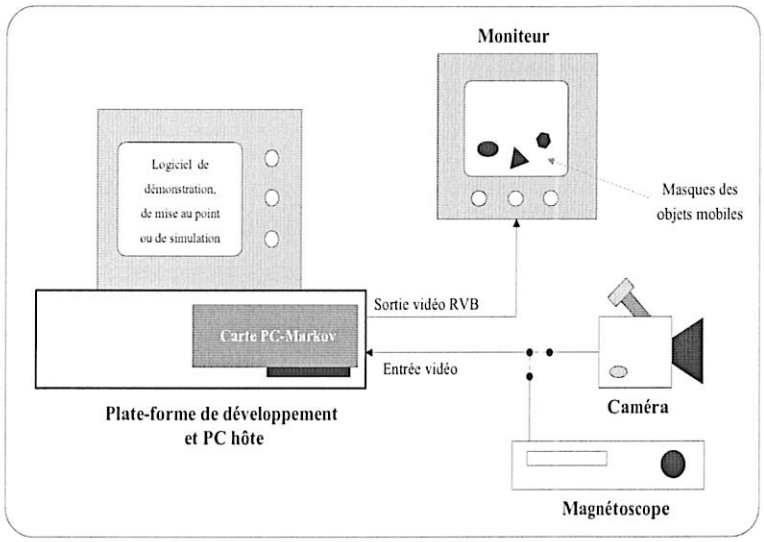

Figure 17: Environnement de test du prototype.

du signal auditif, la vision du mouvement des lèvres du locuteur est donc une donnée précieuse, qui peut être exploitée pour réaliser un système automatique de reconnaissance de la parole, ou de synthèse de visages parlants. Dans ce but, on peut développer un algorithme de détection automatique du mouvement des lèvres d'un locuteur [11]. Un projet qui s'inscrit dans le cadre de la compression audio-visuelle consiste à équiper le locuteur d'un casque léger qui comporte, en plus du microphone, une micro-caméra couleur solidaire du crâne et dirigée vers la zone des lèvres. On reste bien dans le cas d'une caméra fixe par rapport au locuteur et on peut donc appliquer les principes de détection exposés précédemment, en adaptant les observations et le modèle à l'application envisagée.

Les grandes lignes du traitement sont les suivantes : on acquière une séquence couleur RVB (rouge-vert-bleu) du mouvement de la bouche, la région d'intérêt allant de la base du nez au menton. On transforme l'espace RVB en l'espace HIP (teinte ou *hue* en anglais, luminance ou intensité, pureté) qui s'avère mieux adapté au problème

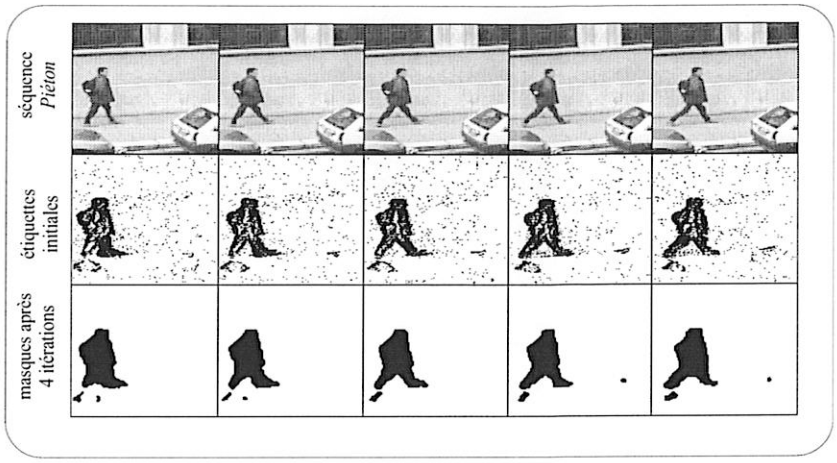

Figure 18: Séquence acquise à la cadence de 25 images/seconde : le mouvement entre deux images est peu important ; la convergence est atteinte après 4 itérations.

:

$$H = \frac{\pi}{2} - \arctan\left(\frac{2R - V - B}{\sqrt{3}(V - B)}\right) + k \tag{13}$$

$$I = \frac{R + V + B}{3} \tag{14}$$

$$P = \frac{R + V + B}{3} - \min(R, V, B) \tag{15}$$

où: $k = 0$ si $V > B$ et $k = \pi$ sinon.
En effet :

- la teinte rouge, qui est prédominante sur les lèvres, est une observation spatiale pertinente,
- la pureté permet de s'affranchir des zones d'ombre (car elle est voisine de zéro dans les zones d'ombre),
- enfin, les variations temporelles de luminance donnent l'information de mouvement.

On utilise deux types d'information de bas niveau :

- une information spatiale sur la teinte rouge (*hue*): $h(s)$,
- une information temporelle sur la différence d'images (*frame difference*) : $fd(s)$.

$$fd(s) = I_t(s) - I_{t-1}(s) \tag{16}$$

$$h(s) = \left[256 - \left(\frac{H(s) - H_m}{\sigma}\right)^2\right] \times 1_{P(s)>\delta} \times 1_{|H(s)-H_m|\leq 16.\sigma} \tag{17}$$

H_m représente la valeur moyenne de la teinte des lèvres (determinée au préalable sur la première image), σ est l'écart-type de la teinte des lèvres (valeur empirique, typ. $4 \leq \sigma \leq 9$) et δ est un seuil appliqué sur la pureté (typ. $50 \leq \delta \leq 100$). La notation $1_{condition}$ dénote une fonction binaire qui vaut 1 si la condition est vraie, 0 sinon.

En pratique, l'algorithme utilise trois observations pour chaque site : $h_{t-1}(s)$, $h_t(s)$ et $fd(s)$. Les Fig. 19 et 20 illustrent ces champs d'observation sur une séquence typique.

Figure 19: *De haut en bas* : séquence des luminances; séquence des observations spatiales h_t à 5 instants t (régions à dominante rouge en blanc, $\delta = 0$).

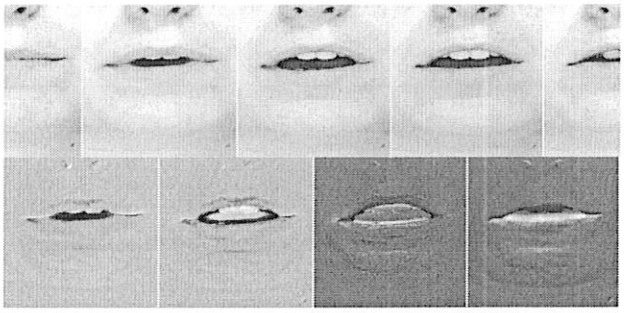

Figure 20: *De haut en bas* : même séquence de luminances; séquence des observations temporelles fd (valeurs positives en blanc, négatives en noir, nulles en gris).

La combinaison de ces trois observations permet de définir un jeu de douze étiquettes (Table 1). L'étiquette c_1 par exemple correspond à la présence de la teinte rouge des lèvres en t, à l'absence de teinte rouge en $t-1$, avec une observation de mouvement détecté positive. On définit alors un modèle énergétique du problème à traiter [11], et on minimise une fonction d'énergie totale faite de cinq termes : trois termes d'attache aux trois observations, un terme d'interaction spatiale et un terme d'interaction temporelle :

$$U = \sum_{s \in S} \Big[\lambda [U_{h_t}(s) + U_{h_{t-1}}(s)] + U_{fd}(s) + U_{sp}(s) + U_{tp}(s) \Big] \tag{18}$$

où λ est un coefficient de pondération sur les énergies d'attache aux observations spatiales. La minimisation de cette fonction d'énergie permet finalement d'extraire de la région d'intérêt les lèvres du locuteur, comme illustré sur la Fig. 21.

Table 1: Observations, codage bas niveau et les 12 étiquettes initiales correspondantes (typ. $\theta = 10$, $\gamma = 100$).

observations			détection initiale en s		codage			étiquettes
$h_t(s)$	$h_{t-1}(s)$	$fd(s)$	teinte	mvt	$r_t(s)$	$r_{t-1}(s)$	$m(s)$	$e(s)$
$< \gamma$	$< \gamma$	$\|.\| < \theta$	$\emptyset$	$\emptyset$	0	0	0	a_0
		$> \theta$		$+$			1	a_1
		$< -\theta$		$-$			2	a_2
	$> \gamma$	$\|.\| < \theta$	$t-1$	$\emptyset$		1	0	b_0
		$> \theta$		$+$			1	b_1
		$< -\theta$		$-$			2	b_2
$> \gamma$	$< \gamma$	$\|.\| < \theta$	t	$\emptyset$	1	0	0	c_0
		$> \theta$		$+$			1	c_1
		$< -\theta$		$-$			2	c_2
	$> \gamma$	$\|.\| < \theta$	t & $t-1$	$\emptyset$		1	0	d_0
		$> \theta$		$+$			1	d_1
		$< -\theta$		$-$			2	d_2

8 Conclusion

On a passé en revue un certain nombre de techniques applicables en détection de mouvement 2D dans les séquences d'images. Mais les outils de traitement présentés ici sont de portée très générale et peuvent être utilisés dans d'autres domaines d'applications ou sur d'autres types de données que des séquences d'images. Ils sont en effet applicables dans tout problème mal posé portant sur des signaux à deux ou trois dimensions, quelle que soit la nature de ces dimensions, et où il est nécessaire d'une manière ou d'une autre de régulariser la solution en introduisant des contraintes ou des connaissances a priori.

References

[1] B. Benmiloud and W. Pieczynski. Estimation des paramètres dans les chaînes de Markov cachées et segmentation d'images. *Traitement du signal*, 12(5):433–454, 1995.

[2] J. Besag. On the statistical analysis of dirty pictures. *J. R. Statist. Soc. B*, 48(3):259–302, 1986.

[3] P. Bouthémy and P. Lalande. Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32(6):1205–1212, June 1993.

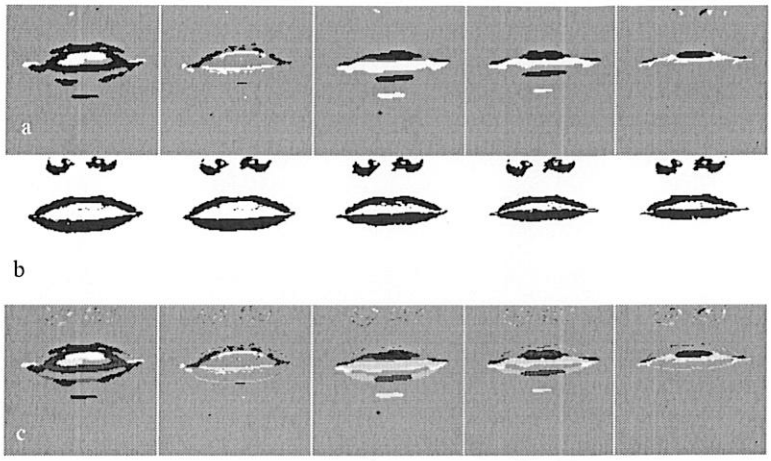

Figure 21: Champs après relaxation. *a)Champs de mouvement M; b)Champs de teinte rouge R_t avec $\delta = 100$; c)Champs E_t des 12 étiquettes représentées en niveaux de gris.*

[4] A. Caplier. *Modèles markoviens de détection de mouvement dans les séquences d'images : Approche spatio-temporelle et mises en oeuvre temps réel.* PhD thesis, Institut National Polytechnique de Grenoble, December 1995.

[5] A. Caplier and F. Luthon. Approche spatio-temporelle pour l'analyse de séquences d'images. Application en détection de mouvement. *Traitement du signal*, 14(2):195–208, 1997.

[6] R. Chellappa and A. Jain, editors. *Markov Random Fields : Theory and Application.* Academic Press, Inc., San Diego, 1993.

[7] J.P. Dérutin, B. Besserer, and J. Gallice. A parallel vision machine : Transvision. In B. Zavidivique and P.L. Wendel, editors, *Proc. of Computer Architecture for Machine Perception*, pages 241–251, Dec. 1991.

[8] C. Dumontier. *Etude et mise en oeuvre temps réel d'un algorithme de détection de mouvement par approche markovienne.* PhD thesis, Institut National Polytechnique de Grenoble, November 1996.

[9] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on PAMI*, 6(6):721–741, November 1984.

[10] Y.Z. Hsu, H.H. Nagel, and G. Reckers. New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics and Image Processing*, 26:73–106, 1984.

[11] F. Luthon and M. Liévin. Lip motion automatic detection. In *10th Scandinavian Conference on Image Analysis*, pages 253–260, Lappeenranta, Finland, June 1997.

[12] F. Luthon, G.V. Popescu, and A. Caplier. An MRF-based motion detection algorithm implemented on analog resistive network. In *3rd European Conference on Computer Vision*, pages 167–174, Stockholm, Sweden, May 1994.

[13] W. Pieczynski. Champs de Markov cachés et estimation conditionnelle itérative. *Traitement du signal*, 11(2):141–153, 1994.