Algorithme cellulaire, itératif et asynchrone d'estimation de mouvement

A Cellular Iterative and Asynchronous Motion Algorithm

par Patricia PLANET

LIME, UJF/ISTG/31, Bt D de Physique
122, rue de la Piscine – Domaine Universitaire, BP 53
F-38041 Grenoble cedex 9

résumé et mots clés

La communication visuelle est en pleine expansion, l'avènement du multimédia et ses besoins en images dynamisent fortement la recherche dans les domaines du codage et de la compression d'images. Il semble que les codeurs futurs seront basés sur des techniques de codage par modèles. Il s'agit à ce niveau essentiellement d'analyse et de synthèse d'images. Cependant toute analyse d'images dépend de primitives utilisant les caractéristiques obtenues par des traitements d'images dits de « bas-niveau » (ou niveau pixel), en particulier l'estimation de mouvement est une primitive de traitement bas-niveau extrèmement importante dans le domaine du codage d'images.

Une première partie de cet article est consacrée à l'étude d'un algorithme cellulaire et itératif d'estimation de mouvement, algorithme développé à partir d'une modélisation markovienne du champs de vecteurs mouvement et d'une méthode de relaxation déterministe. Nous montrons qu'il est possible d'obtenir un champ de vecteurs mouvement apportant un très bon compromis entre la minimisation de l'erreur de prédiction et la cohérence du champ de vecteurs (critères à prendre en compte pour respectivement la réduction de l'information temporelle dans les codeurs classiques, ou l'analyse de scène dans les nouveaux codeurs).

La seconde partie décrit l'étude de cet algorithme d'estimation de mouvement dans un mode de fonctionnement peu ordinaire qui est l'asynchronisme. Ce mode peut être particulièrement intéressant si on vise une implémentation massivement parallèle d'un tel algorithme. L'asynchronisme possède des atouts architecturaux irréfutables et, nous le verrons, des atouts fonctionnels intéressants, montrant ainsi que l'étroite adéquation algorithmes-architectures est très importante et souvent bénéfique.

Traitement d'image, Estimation de mouvement, Algorithme cellulaire itératif, Asynchronisme, Parallélisme.

abstract and key words

The growth of visual communication involves needs for image compression that are more and more important. It seems that futur coders will be based on the structural and semantic relevance of the image contents into play, instead of classical information-theoretic waveform coding.

In this work we apply theoretic concept of Markov modelisation and mean field annealing to the motion analysis problem. Our aim is to obtain a motion estimation algorithm highly parallelisable and dedicated, first, to error prediction minimization and second, to motion analysis problem. We obtain a cellular and iterative algorithm. Such an algorithm can be implemented on array processors with locally mesh connections.

The second part of this work is based on asynchronous iteration. This iteration mode is very suitable for architectural problem. We will see that the convergence of the algorithm in this iteration mode is verified. This part is an illustration of benefits of algorithms and architectures interactions.

Image processing, Motion estimation, Cellular and interative algorithm, Asynchronism, Parallelism.

1. introduction

1.1. contexte

La communication visuelle est l'une des briques du multimédia sans doute la plus importante en terme de volume de données et de besoins de calcul. Le traitement, la transmission et le stockage des images sont alors indissociables de la compression. De façon plus précise, nous nous intéressons ici aux normes de transmission. Les séquences d'images sont caractérisées de manière *externe* et *interne*. La caractérisation externe définit la séquence en terme de taille (nombre de pixels par ligne et nombre de ligne) et de cadence (nombre d'images par seconde). La caractérisation interne, d'une séquence, est définit par la méthode de compression, on obtient alors la norme. La nécessité de regrouper les différentes normes s'est très vite fait sentir et des groupes de travail ont alors été établis pour tenter de trouver une solution à tous ces problèmes. Le groupe MPEG (Moving Picture Expert Group) travaille sur les normes de séquences d'image; il a déjà définit plusieurs normes, MPEG1, MPEG2 et travaille sur les suivantes (en particulier MPEG4). Les normes pour l'instant établies ne définissent en fait que le minimum nécessaire à la compatibilité codeurs–décodeurs et l'objectif à long terme pour les images numériques serait d'intégrer toutes ces normes dans le cadre d'une sorte de méta–norme universelle. Celle–ci devrait permettre une indépendance de la représentation interne de l'image par rapport aux capteurs de saisie des images et aux différents matériels de visualisation des images. C'est dans cette course vers la « standardisation » des normes que ce sont engagés chercheurs et industriels.

1.2. objectifs

Notre travail se situe dans ce contexte du codage d'images. Quel que soit le type de codeur envisagé le traitement d'images au niveau pixel est toujours indispensable. Parmi ces codeurs, nous nous sommes intéressés plus précisément à l'estimation de mouvement par le calcul du flot optique. Ce problème est crucial pour tous les types de codeurs, qu'il s'agisse d'une méthode de prédiction (l'objectif est de minimiser l'erreur de prédiction) ou une primitive d'analyse de scène (l'objectif est de déterminer un champ de vecteur mouvement cohérent avec le mouvement réel). En effet, l'estimation de mouvement effectuée dans le cadre d'un codage d'images prédictif temporel est *a priori* très différente de celle étudiée aux fins d'analyse d'une scène animée. Dans le premier cas c'est la minimisation brute de l'erreur de prédiction inter–images qui est le seul critère à prendre en compte; la cohérence globale du champ de déplacement calculé et sa relation avec le mouvement réel dans la scène sont au contraire déterminants dans le second cas, et l'on a alors recours aux techniques relevant de la vision bas—niveau [Dub92] [Bou 88]. On a expérimenté ici l'application d'une même technique dans ces deux contextes d'utilisation. Un objectif a priori en était de valider dans le cadre d'un schéma de codage hybride classique des algorithmes qui pourront permettre une évolution vers des principes de codage plus avancés (par régions 2D ou modèles 3D) pour lesquels une chaîne complète reste un objectif à long terme. La seconde partie, de cet article décrit l'algorithme développé ainsi que son application dans un codeur de première génération et les perspectives intéressantes pour les codeurs des générations suivantes.

Les traitements de niveau pixel sont généralement hautement parallélisables et il est souvent possible d'envisager des architectures spécialisées dans lesquelles un processeur est affecté au traitement d'un pixel. Ces architectures peuvent être modélisées par les réseaux d'automates cellulaires [Gol90], [Wei89]. L'algorithme d'estimation de mouvement auquel nous aboutissons peut être vu comme un cas particulier d'un algorithme cellulaire et itératif implémentable sur un réseau d'automates cellulaires à connexions locales de type maille carrée, répondant à l'équation générique suivante :

Soit X le vecteur d'état à N dimensions, l'équation générique en chaque site du réseau (ou pour chaque composante x_{ij} du vecteur d'état) est la suivante :

$$x_{ij}^{n+1} = f(x_{ij}^n, x_{i-1,j}^n, x_{i+1,j}^n, x_{i,j-1}^n, x_{i,j+1}^n) \qquad (1)$$

avec $i : 1 \cdots i\max$; $j : 1 \cdots j\max$; $N = i\max \cdot j\max$, où $i\max$ est le nombre de pixel par ligne et j max le nombre de pixels par colonnes, dans le cas d'une image. Chaque site (i, j) localise l'automate dans le réseau, n est le numéro d'itération de l'automate et f est la fonction de mise à jour de chaque automate pour le calcul de sa variable x_{ij}. f est identique pour chaque automate dans le cadre de la définition d'un réseau d'automates cellulaires. Le réseau d'automates cellulaires correspondant à l'équation (1) peut se représenter par la figure 1.

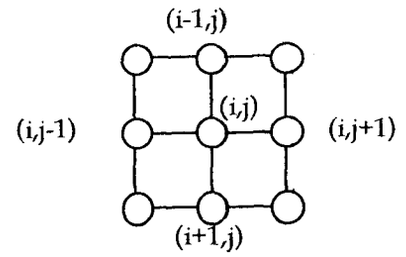

Figure 1. – Réseau d'automates cellulaires à maille carré.

Le second objectif de ce travail est orienté vers les problèmes architecturaux que posent l'implémentation de modèles de réseaux d'automates cellulaires interconnectés localement, sur des architectures spécialisées . Sans entrer trop dans les détails techniques de réalisations VLSI de ces architectures, il est démontré qu'un fonctionnement purement synchrone de ces architectures n'est pas envisageable si l'on vise un parallélisme massif (un

processeur par pixel). Deux contraintes majeures limitent ce parallélisme, il s'agit d'une part des problèmes de dérive d'horloge à distribuer et d'autre part des problèmes liés à la consommation. Une solution à ces problèmes peut être l'asynchronisme. Nous avons donc étudié le fonctionnement asynchrone de cet algorithme d'estimation de mouvement, cherchant à profiter fonctionnellement et structurellement de l'asynchronisme. Les résultats sur cet algorithme sont présentés au § 3 et peuvent être généralisés à de nombreux algorithmes cellulaires et itératifs.

2. algorithme

2.1. identification du problème au modèle énergétique

La modélisation markovienne, en tant qu'apparentée aux techniques de régularisation, a été utilisée à l'origine pour des problèmes de restauration d'images [Gem 84], et appliquée ultérieurement à la modélisation de champs de déplacement en analyse d'images [Kon 88], [Hei 92]. Les contraintes de régularité des champs solutions sont imposées par le biais d'un voisinage spatio–temporel pour prendre en compte à la fois l'objectif de lissage temporel inhérent à la prédiction causale et le lissage spatial lié à la cohérence structurelle du champ.

Sans développer les calculs des différentes étapes de la modélisation markovienne, dont les détails peuvent être trouvés dans [PLa–2 94], on rappelle ici les principales hypothèses de cette modélisation :

– Choix du voisinage

Chaque pixel possède un voisinage, nous l'avons choisi du premier ordre :

Le voisinage du pixel (i, j, n) est noté
$N_{ij} = \{(i-1, j, n); (i+1, j, n); (i, j-1, n);$
$(i, j+1, n); (i, j, n-1)\}$.

– Connaissance a priori

Soit $\vec{d_{ij}} = (d_{xij}, d_{yij})$ le vecteur mouvement, variable 2D, en chaque pixel (i, j)
il faut choisir une contrainte a priori. L'une des plus simples est de dire que les pixels voisins doivent avoir des vecteurs mouvement semblables. Cette hypothèse n'est en théorie valable qu'en dehors des frontières. Ici, classiquement, il faudrait introduire des processus de lignes dans le but d'inhiber la contrainte de lissage du champ sur les discontinuités de l'image (contours des objets), nous verrons comment on les introduits implicitement dans le calcul des dérivées spatiales. Les contraintes étant choisies, l'équivalence de Gibbs permet alors d'écrire la probabilité a priori :

$$P(\vec{D} = \vec{d}) = \frac{1}{Z} \exp\left(-\frac{E_c(\vec{d})}{T}\right)$$

avec $$E_c = \sum_{ij} V_c(i,j) = \sum_{ij} \sum_{kl \in N_{ij}} \|\vec{d_{ij}} - \vec{d_{kl}}\|^2$$

où Z est une constante normalisatrice et T un paramètre de pondération des contraintes appelé température par analogie avec un système de bain de chaleur en mécanique statistique.

– Connaissance a posteriori

On commence par choisir une observation du champ à estimer : une observation simple peut être le champ de vecteurs mouvements issus de la dfd (displaced frame difference) développée au premier ordre (l'utilisation d'un ordre supérieure serait illusoire étant donné le peu de validité des dérivées au niveau des frontières). Ensuite s'inspirant de ce qui a été fait en restauration d'images, et en supposant que le champ de vecteur mouvement réel est le champ observé et bruité par un bruit gaussien deux dimensions, on peut retrouver l'expression connue introduite par Konrad et Dubois [Kon 88] :

$$p(\vec{d}/obs.) = \frac{1}{Z} \exp\left(\frac{-1}{T} \sum_{ij} \left(T(I_x d_x + I_y d_y + I_t)^2 + \sum_{kl \in N_{ij}} \|\vec{d_{ij}} - \vec{d_{kl}}\|^2 \right)\right) \quad (2)$$

où $I_{xij}, I_{yij}, I_{tij}$ sont respectivement les estimations des dérivées au pixel (i, j) de l'intensité, par rapport à l'axe des i, à l'axe des j, puis l'axe du temps (différence inter–image). Par la suite quand la confusion ne sera pas possible nous omettrons volontairement pour la clarté des formules les indice (i, j).

Il est en théorie très facile d'introduire ces processus de lignes dans le modèle. Cependant étant donné que nous recherchons un algorithme simple en vue d'une implémentation matérielle, nous avons préféré utiliser l'observation suivante :

– il est possible de s'affranchir des processus de lignes en choisissant en chaque pixel la dérivée spatiale, dans chacune des deux directions, comme celle des deux dérivées partielles à droite et à gauche qui possède la plus faible pente.
Par exemple : $I_{xij} = \min(I^+_{xij}, I^-_{xij})$ avec $I^+_{xij} = I_{i+1,j} - I_{ij}$ et $I^-_{xij} = I_{i,j} - I_{i-1j}$, sachant que l'image a été préalablement filtrée par un passe–bas, tentant ainsi de rendre dérivable une fonction qui ne l'est pas, en particulier aux frontières des objets.

Intuitivement on associe ce mode de calcul des dérivées spatiales aux processus de lignes car ainsi la dérivée d'un pixel appartenant au bord d'un objet n'exprime pas le contour de l'objet mais bien la fin de l'objet. C'est une démarche inverse à ce qu'on ferait pour une détection de contours, qui nous a permis d'obtenir à la fois une meilleure cohérence sur le champ de mouvement et une réduction de l'erreur de prédiction, des résultats ont été montrés dans [Pla294].

Cette modélisation permet donc de ramener par le biais de l'équivalence de Gibbs le problème initial à la minimisation d'une

fonctionnelle énergétique dont l'expression est :

$$E = \sum_{xy} \left[T(I_x d_x + I_y d_y + I_t)^2 + \sum_{kl \in N_{xy}} \|d_{xy} - d_{kl}\|^2 \right] \quad (3)$$

Le premier terme correspond au lien entre les observations (les gradients de l'intensité I), et les composantes du déplacement **d**, tandis que le deuxième terme représente la contrainte de régularité du champ de déplacement sur le voisinage N_{ij} considéré en chaque pixel de coordonnées (i, j).

L'utilisation de l'équation de la DFD suppose que nous traitons de faibles mouvements, cependant il est clair que pour rester le plus général possible cet algorithme sera ensuite évalué avec une étude multirésolution, ceci nous permettant de traiter des mouvements de plus amples amplitudes.

2.2. stratégie de minimisation par relaxation déterministe

Les algorithmes classiques de relaxation stochastique (recuit simulé et échantillonneur de Gibbs) sont optimaux pour la résolution de tels problèmes identifiés à la minimisation d'énergie d'un système physique, mais leur lourdeur en rend l'utilisation peu praticable. De nombreux auteurs [Pet 89], [Hér 91], [Bil 90], [Zha92] [Zer 90] se sont intéressés à des algorithmes déterministes inspirés de la théorie du champ moyen en physique statistique et simulant le comportement du recuit stochastique. A une température T donnée, le recuit simulé fait évoluer le système jusqu'à un quasi–équilibre thermodynamique. L'algorithme du recuit en champ moyen (MFA) consiste à simuler le comportement d'un système soumis à un recuit de façon déterministe, en écrivant que : $\langle (E(\mathbf{d}) \rangle = E_{eq}(T)$.

Les calculs étant développés dans [Pla2 94], nous ne donnons ici que le résultat. Les équations itératives permettant d'obtenir la solution s'écrivent :

$$d_x^{n+1} = \frac{-I_x\, T(I_t + I_y\, d_y^n) + \sum_{kl \in N_{xy}} d_{x_{kl}}^n}{T\, I_x^2 + 5} \quad (4)$$

(la constante du dénominateur, ici 5, correspond au nombre de voisins de chaque pixel).

Une équation similaire étant obtenue pour la composante en y.

Le compromis entre minimisation de l'erreur de prédiction et cohérence du champ peut être en parti réglé par le choix de la valeur initiale du paramètre T. Cet algorithme itératif prend comme valeur initiale les observations. Classiquement le paramètre T diminue au cours des itérations. Pour un algorithme déterministe il a été montré qu'une loi de descente du type $T^{n+1} = T^n * 0,975$ donnait le comportement souhaité à l'algorithme [Her 91]. Pour mémoire Geman et Geman [Gem 84] ont montré que la loi de descente optimale dans le cas d'un recuit stochastique est une loi de type logarithmique. La valeur initiale de T est quant à elle déterminée de façon empirique (nous avons choisit $T^0 = 0,1$).

Cet algorithme, on le rappelle, a pour but d'offrir un bon compris entre méthode de prédiction et cohérence du champ de vecteurs mouvement. Nous montrons après avoir décrit le type de codeur utilisé, les différentes performances de cet algorithme face à ces objectifs a priori incompatibles.

2.3. schéma causal de compensation de mouvement

Les techniques classiques d'appariement de blocs (« block–matching ») sont bien adaptées à des schémas de prédiction inter–image de type non–causal (antérogrades), où l'on doit transmettre l'information de mouvement avec l'erreur de prédiction (figure 2–b). Le terme non–causal est utilisé, car pour prédire l'image $\hat{I}_n$ on utilise l'image réelle I_n. L'estimation de mouvement est un problème d'optimisation parfaitement déterminé dans ce cadre purement prédictif, contrairement à l'analyse d'un champ de mouvement dense cohérent par rapport aux mouvements réels de la scène ou de la caméra, qui est classiquement un problème sous-déterminé d'optique inverse. Un champ dense peut également être utilisé comme prédicteur inter–images, mais le codage du champ risquant dans ce cas d'annuler le bénéfice d'une estimation plus précise [Nic 90], on est conduit à s'intéresser à des schémas de type causal ou « rétrograde » [Dri 90] ne nécessitant pas la transmission de vecteurs mouvement (figure 2–a). Le principe en est d'utiliser les images $n-1$ et $n-2$ reconstruites à l'identique au codeur et au décodeur pour prédire l'image n. Il est entendu que l'utilisation d'une prédiction causale n'est possible que durant une scène à mouvement continu et quand l'erreur de prédiction devient trop forte (par exemple supérieure à une simple différence inter–images) un raffraichissement d'images est alors effectué.

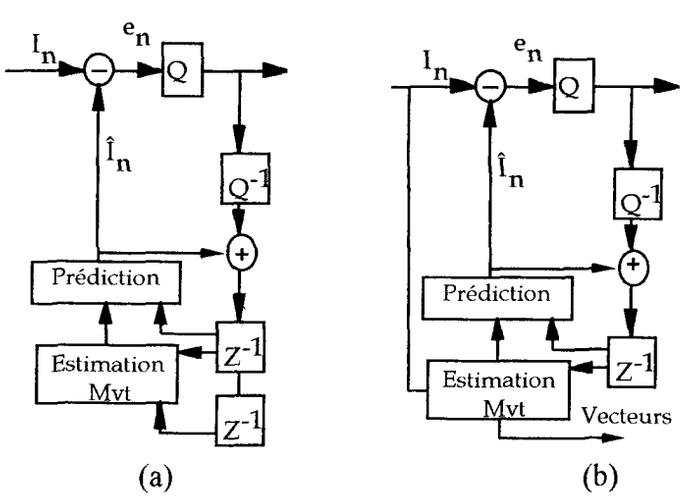

Figure 2. – Boucle prédictive inter-images causale/retrograde (a) et non-causale/antérograde (b).

où : Q est un opérateur de quantification vectorielle.

Z^{-1} est une mémoire d'image.

2.4. évaluations

2.4.1. boucle de codage sans perte

Ces algorithmes appliqués à l'estimation de mouvement ont été évalués dans le cadre d'un schéma de codage prédictif sans pertes afin d'isoler les performances de l'estimation de mouvement elle–même de celles des autres composantes de l'algorithme.

Sur deux séquences (Train–calendrier, avec un ensemble de mouvements multiples (mouvement de caméra + mouvement d'objets), et Père Noël, avec un mouvement simple et localisé), les algorithmes suivants ont été testés avec les résultats (en entropie par pixel) donnés dans le tableau ci–dessous :

1. Différence inter–image directe sans compensation de mouvement : **Diff_Im**

2. Algorithme classique d'appariement de blocs, avec fenêtre de recherche $\pm$ 7 et blocs 16 $\times$ 16, en prenant en compte l'entropie de l'ensemble des vecteurs mouvement associés à chaque bloc : **B_16.16**

3. Idem, avec uniquement l'entropie de l'erreur : **Err_B_16.16**

4. Algorithme classique de Horn et Schunk [Hor 81] dans une boucle inter–images causale : **H_S**

Les 3 systèmes suivants utilisent les algorithmes globalement récursifs proposés. Une variante avec recuit de l'algorithme ICM [Bes 86] et le recuit en champ moyen (MFA) ont été tous deux testés, avec des résultats équivalents en termes de performance. A noter, que le nombre d'opérations élémentaires pour l'ICM est plus important que celui du MFA.

5. Globalement récursif avec prédiction causale : **MFA**

6. Le même que 5, basé sur représentation multirésolution (pyramide des moyennes) : **Pyr_MFA**

7. Algorithme globalement récursif sur pyramide inséré dans un schéma non–causal. L'entropie donnée ne prend pas en compte le champ de mouvement; elle doit être comparée à celle obtenue en 3 : **Pyr_MFA_nc**

On voit que les techniques proposées améliorent la qualité de la prédiction dans le cadre d'un schéma causal par rapport à des techniques plus simples comme Horn et Schunk (H_S). Le fait d'avoir un champ plus cohérent spatialement et temporellement grâce aux contraintes introduites dans la modélisation markoviennne se traduit donc à la fois par une meilleure cohérence du champ par rapport à la structure géométrique de l'image, ce qui est apparent quand on le visualise, mais aussi par une meilleure prédiction dans le cadre d'un schéma de codage différentiel inter–images de type causal.

Tableau 1. – Entropie en bits/pixel des différentes méthodes, sur deux séquences tests.

Algorithmes	Train–calendrier	Père–Noël
Diff	5.09	4.42
B_16.16	4.46	4.256
Err_B_16.16	4.44	4.254
H_S	4.52	4.46
MFA	4.44	4.27
Pyr_MFA	4.34	4.28
Pyr_MFA_nc	4.27	4.19

La multirésolution utilisée ici, bien que très simple, apporte de meilleurs résultats que la mono–résolution, en particulier dans les séquences où les mouvements sont relativement importants. La pyramide peut combler un déficit de performances dû à un choix de voisinage trop simple. Dans le cas de la séquence du « Père Noël », où le mouvement est faible et localisé dans une petite zone, la pyramide n'apporte rien de mieux par rapport à l'algorithme mono–résolution. Le rôle de la pyramide est donc bien dirigé sur l'estimation du mouvement de plus forte amplitude, ce qui ne fait que confirmer les résultats obtenus par, entre autres, Perez et Heitz [Hei 92].

2.4.2. codeur complet

L'évaluation entropique de l'algorithme vue précédemment justifie qu'il soit testé dans un schéma complet de codage. Le codeur utilisé est basé sur une décomposition en sous–bandes à l'aide de filtres QMF. La compensation de mouvement s'effectue dans un premier temps sur la basse fréquence puis sur l'image de pleine résolution. L'erreur de chacune des sous–bandes est ensuite quantifiée vectoriellement. Nous présentons ci–dessous les différents rapports signal sur bruit obtenus pour différents taux de compression. Ces chiffres sont les moyennes effectuées sur dix images.

Tableau 2. – Rapport signal sur bruit en fonction du taux de compression

compression	15	20	25	30	35	40	80
SNR moyen	35	33,6	32,5	31,6	31	30,3	28

Ces valeurs moyennes masquent le phénomène de divergence de l'algorithme dû à la causalité du codeur. L'algorithme donne un très bon résultat sur la première image puis la qualité baisse un