Agrégation, sélection et utilisation de l'information de mouvement issue d'un flux MPEG

Renan COUDRAY, Bernard BESSERER,

Laboratoire Informatique Image Interaction, Université de La Rochelle
Av. Michel Crépeau, 17042 La Rochelle Cedex 1, France
renan.coudray@univ-lr.fr.fr, bernard.besserer@univ-lr.fr.fr,
tel:(33) 05 46 45 83 22, fax:(33) 05 46 45 82 42

Résumé – Le stockage d'une grande quantité de vidéos numériques, ainsi que leur diffusion, a été rendu possible grâce à l'utilisation de techniques de compression. Le standard MPEG, largement utilisé, atteint un taux de compression important en réduisant la redondance temporelle. Pour cela, des informations de compensation de mouvement sont calculées par l'encodeur et sont transmises dans le flux. Or, de nombreuses applications (segmentation, indexation, ...) requièrent une information relative au mouvement apparent d'une séquence d'image. Nous présentons dans cet article une façon de réorganiser les données présentes dans le flux MPEG afin de pouvoir réutiliser, avec une fiabilité accrue, l'information de mouvement. La charge calculatoire est moindre, car l'estimation de mouvement est effectuée à l'encodage. A titre d'exemple, nous montrons comment le flux MPEG et les informations qui y sont contenues peuvent être utilisées pour estimer le mouvement accompli par la camera (*global motion*).

Abstract –
Broadcast of digital video became possible using compression methods. MPEG standard is widely used because it allows large ratio compression by canceling redundant temporal information. MPEG compression is performed by a codec that generates information such as motion compensation field. Many applications (segmentation, indexing, ...), which process image sequences, rely on flow field to figure out apparent movement. In this article, we present a method which use the motion compensation field and select only valid information to improve its reliability. Applications gain better performance because motion estimation is carried out by MPEG codec at encoding step. For example, we demonstrate how MPEG flow can be efficiently used to extract global camera movements.

1 Introduction

De nombreuses applications impliquant la manipulation des objets vidéo comme l'indexation ou la segmentation ont besoin de calculer le mouvement présent dans une séquence d'images. Cette opération étant souvent coûteuse en temps de calcul, on se propose de réutiliser l'information de compensation de mouvement présente dans les formats de compression comme le MPEG, information estimée et enregistrée lors de l'encodage du flux vidéo. Le standard MPEG [1](Motion Picture Expert Group), le plus répandu actuellement, est notamment utilisé par la norme DVB (Digital Video Broadcasting, [2]), norme de diffusion des télévisions numériques, ouvrant un champ important d'applications potentielles.

Nous Présentons deux volets de notre recherche : La mise en forme et la sélection d'informations pertinentes de mouvement à partir du flux MPEG (en utilisant autant que possible les informations sous leur forme compressée) et un exemple d'utilisation de ces informations pour estimer le mouvement de camera (*global motion*).

2 Mise en forme des vecteurs mouvements du MPEG

Il est fondamental de rappeler que l'information de compensation de mouvement enregistrée dans un flux MPEG n'est pas forcement valide au sens du flot optique (déplacement des objets). Il s'agit de vecteurs pointant des blocs similaires, le but de cette information étant d'améliorer le taux de compression en réduisant les redondances temporelles. La compensation de l'image s'effectue par bloc (découpage de l'image en macro-blocs). Si ce bloc contient une structure, des contours, alors la probabilité d'estimer un mouvement (au sens du flot optique) augmente, du moins pour la composante orthogonale aux structures présentes dans le bloc. A l'inverse, une région uniforme peut être appariée avec n'importe quelle autre région uniforme ; les vecteurs de compensation affectés à ce bloc ne correspondent probablement pas au mouvement réel.

Nous choisissons de combiner les vecteurs mouvements présents dans les P et les B-frames (images compensées) afin d'obtenir un vecteur pour chaque macro-bloc d'une I-frame (notre information représente alors le mouvement d'une I-frame à une autre, soit les mouvements perçus le long d'un GOP (Group Of Picture)). Ces informations pourrons être rejetées sur la base de l'analyse des coefficients DCT (Discret Cosine Transform), selon la méthode décrite ci-après (2.2).

2.1 Combinaison des vecteurs

Le principe est de chaîner les vecteurs mouvements transmis avec les B et P-frames afin de relier les I-frames. Nous posons comme hypothèse d'être en présence de mouvements du type affine. Nous montrons dans [3] comment combiner et lier bout à bout les vecteurs mouvements, comme illustré sommairement dans **Fig. 1**.

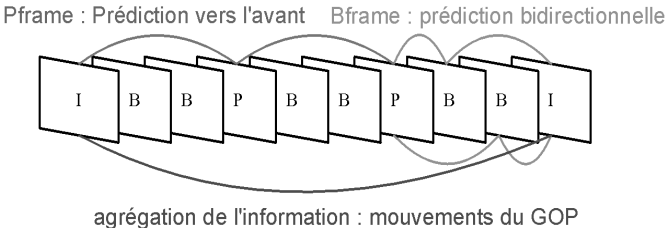

FIG. 1 – Calcul du mouvement du GOP

2.2 Rejet

L'opération de rejet est essentielle pour notre approche, or, la transformation DCT est calculée pour chaque bloc MPEG alors que la compensation de mouvement est calculée par macro-bloc (ensemble de 4 blocs). Afin de se prononcer sur la validité du vecteur mouvement associé au macro-bloc, il faut d'abord rassembler l'information des 4 blocs sous-jacents. Sans cette opération, les variations existantes d'un bloc à l'autre ne seraient pas prises en compte. Par exemple, si nos quatre blocs sont uniformes mais contrastés entre eux, nous ne pouvons pas détecter directement la présence de cette différence qui a pourtant joué un rôle lors de la compensation (opération d'appariement de bloc, ou *block matching*, effectuée par l'encodeur). Après combinaison de l'information, un indice de texture sera calculé, permettant d'estimer la fiabilité du vecteur mouvement issu de la compensation.

2.2.1 Sous échantillonnage

Nous nous appuyons, pour cette première étape, sur une méthode décrite dans [4] qui permet de sous-échantillonner les données DCT, c'est à dire de créer un seul bloc représentant l'information contenue à l'origine dans 4 blocs connexes.

Soit les constantes suivantes (dans le cas du MPEG $N = 8$) :

$$T^N(l,n) = \begin{cases} \frac{1}{\sqrt{N}} \; si \; l = 0 \\ \sqrt{\frac{2}{N}} \cos \frac{(2n+1)l\pi}{2N} \; sinon \end{cases} \tag{1}$$

$$[T_L|T_R] = T^N \quad T_s = T^{N/2} \tag{2}$$

$$C = \frac{1}{\sqrt{2}} \frac{1}{2} ((T_L T_s^t) + (T_R T_s^t)) \tag{3}$$

$$D = \frac{1}{\sqrt{2}} \frac{1}{2} ((T_L T_s^t) - (T_R T_s^t)) \tag{4}$$

T_N et T_s sont des matrices carrées de dimension N et $N/2$. T_L et T_R sont respectivement les parties gauche et droite T_N ; elles ont donc une taille de $(N, N/2)$.

En notant B_i les parties supérieures gauches de dimension $(N/2, N/2)$ des 4 blocs DCT d'un macro-bloc (numérotés de gauche à droite et de haut en bas) et B le résultat de la combinaison (de taille (N, N)), alors les calculs à effectuer sont les suivants :

– dans l'article [4] :

$$\begin{aligned} X &= C(B_1 + B_3) + D(B_1 - B_3) \\ Y &= C(B_2 + B_4) + D(B_2 - B_4) \\ B &= (X + Y)C^t + (X - Y)D^t \end{aligned} \tag{5}$$

– ou de manière équivalente :

$$\begin{aligned} X &= (B_1 + B_2)C^t + (B_1 - B_2)D^t \\ Y &= (B_3 + B_4)C^t + (B_3 - B_4)D^t \\ B &= C(X + Y) + D(X - Y) \end{aligned} \tag{6}$$

Il reste à calculer une valeur proportionnelle à la quantité de "formes" présentes dans B, pour estimer la confiance que l'on peut attribuer au vecteur mouvement attaché à notre macro-bloc.

2.2.2 Texture DCT

Ce que nous dénommons texture ou forme correspond à la variance des niveaux de gris présents dans le macro-bloc. Les valeurs des pixels d'un bloc sont décrites grâce aux coefficients DCT (décomposition fréquentielle) et ceux-ci ne sont pas invariants au changement de position de ces formes. Les travaux décrits dans [6] ont pour but d'indexer des blocs DCT et présentent des indices de texture statistiquement stables aux décalages. Nous proposons de modifier deux de ces indices en retirant la composante continue pour obtenir deux indices t_x, t_y qui représentent respectivement la présence de formes horizontales ou verticales :

$$t_x(B) = \sum_{i=1}^{N} B(0,i)^2 \tag{7}$$

$$t_y(B) = \sum_{i=1}^{N} B(i,0)^2 \tag{8}$$

2.2.3 Combinaison des techniques

L'enchaînement du sous échantillonage et de l'extraction des indices de forme en restant dans le domaine compressé entraîne énormément de calculs. Nous avons étudié la possibilité d'obtenir $t_x(B)$ et $t_y(B)$ directement à partir des valeurs des B_i sans passer par X et Y. La simplification des équations **Eq. (9)** et **Eq. (10)** fut notre point de départ. Une démonstration exhaustive, par l'utilisation successives de règles trigonométriques, conduisant à ces formules serait trop imposante dans le cadre de cet article.

$$t_x(X) = \sum_{n=1}^{N/2} X(0,n)^2 = ... = \frac{t_x(B_1 + B_3)}{4} \tag{9}$$

$$\begin{aligned} t_y(X) &= \sum_{l=1}^{N} X(l,0)^2 = ... \\ &= \frac{t_y(B_1) + t_y(B_3)}{2} + \frac{(B_1(0,0) - B_3(0,0))^2}{4} \end{aligned} \tag{10}$$

On trouve des équations équivalentes pour Y (remplacer B_1 par B_2, et B_3 par B_4). Pour fusionner X et Y les équations sont inversées entre t_x et t_y :

$$t_x(B) = \frac{t_x(X) + t_x(Y)}{2} + \frac{(X(0,0) - Y(0,0))^2}{4} \tag{11}$$

$$t_y(B) = \frac{t_y(X + Y)}{4} \tag{12}$$

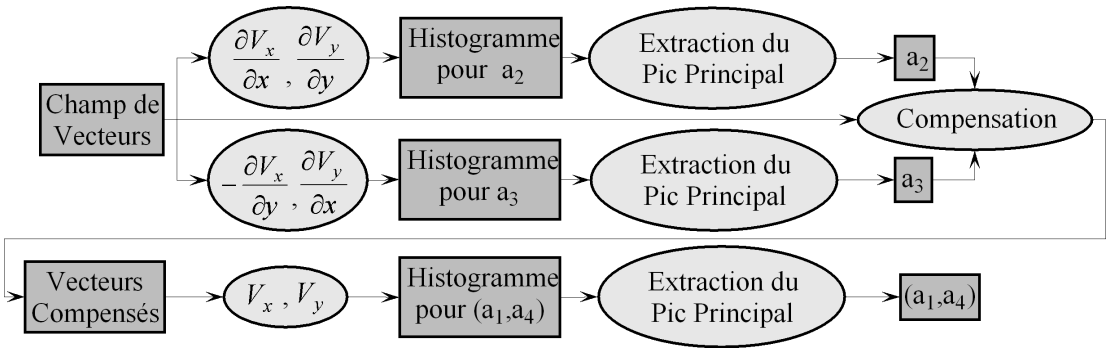

FIG. 2 – Estimation du mouvement global : Synoptique de la méthode basée sur le modèle affine simplifié

Sachant que la composante continue d'un bloc reconstruit est la moyenne des composantes continues des blocs fusionnés et en utilisant les équations **Eq. (9)** et **Eq. (10)**, on obtient t_x en fonction des B_i :

$$t_x(B) = \frac{\frac{t_x(B_1+B_3)}{4} + \frac{t_x(B_2+B_4)}{4}}{2} + \frac{\left(\frac{B_1(0,0)+B_3(0,0)}{2} - \frac{B_2(0,0)+B_4(0,0)}{2}\right)^2}{4} \quad (13)$$

Pour t_y, on ne peut pas connaître la valeur de $t_y(X+Y)$ sans passer par le calcul de X et Y, que l'on veut éviter. Il faut donc se placer dans le cas équivalent (**Eq. (6)**), c'est à dire regrouper les blocs horizontalement en premier pour obtenir au final :

$$t_y(B) = \frac{\frac{t_y(B_1+B_2)}{4} + \frac{t_y(B_3+B_4)}{4}}{2} + \frac{\left(\frac{B_1(0,0)+B_2(0,0)}{2} - \frac{B_3(0,0)+B_4(0,0)}{2}\right)^2}{4} \quad (14)$$

Dans le cas de vidéos entrelacées, les lignes paires et impaires sont souvent encodées séparément en faisant des regroupements (les lignes paires dans deux blocs DCT et les lignes impaires dans les deux autres). Dans ce cas, nous avons 2 vecteurs mouvements différents (les lignes paires et impaires sont compensées de manières indépendantes), mais nous affectons le même indice de texture à ces deux vecteurs. L'influence de la démarcation artificielle introduite pas le regroupement séparé des lignes paires et impaires peut être annulée en retirant l'énergie due aux composantes continues pour l'indice de texture verticale (suppression du terme de droite de la somme de l'équation **Eq. (14)**).

Ces calculs produisent 2 indices de textures qui sont associés aux composante x, y de nos vecteurs mouvements. Ces indices sont transformés en une valeur de fiabilité comprise entre 0 et 1 (utilisation de *least-power influence function*, [11]) permettant de pondérer la qualité de l'information de mouvement selon l'application envisagée. Si la fiabilité d'un vecteur est demandée (et non la fiabilité d'une de ses composantes), le minimum de la fiabilité de ses deux composantes est alors utilisé.

3 Estimation du mouvement global

3.1 Méthode

Les données disponibles pour l'estimation du mouvement global correspondent à un champ de vecteurs épars, et, malgré notre technique de rejet, des vecteurs aberrants subsistent. De nombreuses méthodes d'estimation du mouvement utilisent une technique du type "moindres carrés" [10, 9]. Ces méthodes sont coûteuses en temps de calcul, surtout dans leurs versions récursives permettant le rejet des mouvements des objets. Dans [3] nous proposons une approche d'accumulation dans un espace paramétrique (accumulation des informations proportionnellement à l'indice de fiabilité précédemment présenté), permettant d'écarter les vecteurs correspondants aux mouvements des objets présents dans la vidéo (exemple des footballeurs dans **Fig. 3**) ainsi que les vecteurs erronés restants. La **Fig. 2** représente le processus d'estimation du mouvement de la camera, mouvement respectant le modèle affine simplifié (**Eq. (15)**, 4 paramètres). Nous avons étendu la méthode pour le modèle affine (**Eq. (16)**, 6 paramètres).

$$V = \begin{pmatrix} a_1 + a_2x - a_3y \\ a_4 + a_3x + a_2y \end{pmatrix} \quad (15)$$

$$V = \begin{pmatrix} a_1 + a_2x + a_3y \\ a_4 + a_5x + a_6y \end{pmatrix} \quad (16)$$

3.2 Résultats

De nombreuses applications ayant besoin d'une information sur le mouvement présent dans une vidéo peuvent s'appuyer sur ces travaux, en s'accommodant toutefois des restrictions du MPEG et de la méthode : champ de vecteurs épars, un seul champ de vecteurs par GOP seulement et une qualité d'estimation différente selon l'encodeur MPEG utilisé. Nos essais démontrent que les vidéos diffusées sur les chaînes numériques ou provenant de cartes d'acquisition temps réel possèdent une qualité d'estimation suffisante permettant d'imaginer de nombreux cas concrets d'application. Dans [8] des *storyboards* sont crées de manière automatique grâce au calcul du mouvement de la camera. Dans [5] des séquences de sport sont classées par genre grâce au mouvement présent dans les vidéos. Ces deux exemples d'application utilisent les séquences décompressées mais notre technique d'estimation du mouvement conviendrait bien à ceux-ci.

FIG. 3 – Résultat d'une application créant automatiquement une mosaïque à partir d'une séquence (retransmission d'un match de football) au format MPEG. Les I-frames successives sont ajustées entre elles grâce à l'estimation du mouvement de la camera (panoramiques et zooms dans cette sequence).

L'estimation du mouvement caméra ou *global motion* représente l'application générique et simple de notre approche, pouvant être confrontée avec d'autres techniques. Dans [3], nous comparons nos résultats de l'estimation avec une méthode du domaine non compressé, mettant en évidence des résultats très similaires, avec des coûts de calculs réduits (3 à 4 fois le temps réel, sur un pentium 4 2.4 GHz). La **Fig. 3** montre le résultat d'un *mosaicing*, l'arrangement des images étant basé sur l'estimation du mouvement caméra délivrée par cette méthode. Seule la composante continue de chaque bloc est décodée, ce qui explique la qualité de l'image. Ce résultat montre que le mouvement effectué par la camera est correctement estimé par notre méthode. Il reste des imperfections, pour obtenir une mosaïque de qualité supérieure il faudra faire appel aux données des images pour affiner le résultat.

La réutilisation des informations de compensation de mouvement d'un flux MPEG est aussi étudiée pour générer une signature vidéo, destinée à une reconnaissance des contenus diffusés. Des tests sont en cours, notamment la mise en place d'un protocole de sélection d'indices extraits du champ de vecteurs mouvement, protocole adapté à notre problème d'identification de séquences d'images.

4 Conclusion

Nous avons présenté une méthode qui utilise exclusivement les données compressées d'un flux MPEG et qui permet d'obtenir (en temps réel) une information de mouvement exploitable. Malgré les restrictions du MPEG, nos méthodes d'agrégation des vecteurs et leurs rejets éventuels sur la base d'indices de texture, fiabilisent l'utilisation ultérieure de ce champ de vecteurs mouvements. Nos méthodes actuelles délivrent un champ de vecteurs par GOP ; si une précision temporelle plus importante est demandée, il est probable qu'un traitement des données exclusivement dans le domaine compressé ne suffise plus. Malgré ces restrictions, le potentiel recherche est riche et les applications possibles sont nombreuses.

Références

[1] *ISO/IEC 13818-1 and ISO/IEC 13818-2*. 2000.

[2] H. Benoit. *Digital Television MPEG-1, MPEG-2 and Principles of the DVB System, second edition*. Focal Press, 2002.

[3] R. Coudray and B. Besserer. Global motion estimation for MPEG-encoded streams. *Proc. of IEEE Int. Conf. on Image Processing*, 2004.

[4] R. Dugad and N. Ahuja. A fast scheme for image size change in the compressed domain. *IEEE Trans. Circuits Syst. Video Technol.*, 11 :461–474, April 2001.

[5] R. Fablet, P. Bouthemy, and P. Perez. Non parametric motion characterization using temporal gibbs models for content-based video indexing and retrieval. *IEEE Trans. Image Processing*, accepted for publication, 2003.

[6] R. E. Frye and R. S. Ledley. Texture discrimination using discrete cosine transformation shift-insensitive (DCTSIS) descriptors. *Pattern Recognition*, 33(10) :1585–1598, 2000.

[7] J. Heuer and A. Kaup. Global motion estimation in image sequences using robust motion vector field segmentation. *ACM Multimedia*, pages 261–264, November 1999.

[8] S. Porter, M. Mirmehdi, and B. Thomas. Video indexing using motion estimation. *The British Machine Vision Conference*, 2003.

[9] A Smolic and J-R Ohm. Robust global motion estimation using a simplified m-estimator approach. *ICIP*, 2000.

[10] R. Wang and T. Huang. Fast camera motion analysis in MPEG domain. *IEEE Trans. Image Processing*, 3 :691–694, 1999.

[11] Z. Zhang. Parameter estimation techniques : A tutorial with application to conic fitting. *Image and Vision Computing Journal*, 15(1) :56–76, 1997.