

Estimation de mouvement par correspondance de blocs avec fenêtre de recherche angulaire et segmentation de blocs

La plupart des normes de codage vidéo comme la CCITT/H.261 [1] et l'ISO/MPEG2 [2] adoptent les méthodes de correspondance de blocs [3]. Parmi l'ensemble de ces techniques, la méthode de recherche exhaustive (MRE) s'est montrée la plus performante. Néanmoins, elle est très coûteuse en temps de calcul. Par conséquent, plusieurs algorithmes rapides ont été développés [4][5][6]. Ils sont tous basés sur l'hypothèse selon laquelle l'erreur de correspondance décroît d'une façon monotone en se rapprochant de la solution optimale globale. Néanmoins, ces algorithmes peuvent converger vers un minimum local ne correspondant pas à la solution optimale obtenue par la MRE. Par conséquent, l'utilisation de ces algorithmes conduit généralement à une performance moins bonne ou bien à un débit binaire plus important. De ce fait, plusieurs algorithmes ont été proposés [7][8][9] dont l'objectif est de réduire les défauts dus aux minima locaux tout en conservant un temps de calcul plus rapide que celui de la MRE. Nous proposons dans cet article une autre version d'algorithme basé en premier sur l'estimation de mouvement selon des directions bien privilégiées permettant d'avoir une idée a priori sur la direction du mouvement. Ensuite une fenêtre de recherche angulaire est définie en deux étapes autour de la direction de chaque vecteur de mouvement pour trouver enfin le déplacement de chaque bloc. Cet algorithme permet un gain de temps d'environ 3 à 7 fois selon la taille de la fenêtre de recherche angulaire utilisée par rapport à la MRE tout en réduisant le problème du minimum local.

Par ailleurs, l'hypothèse implicite de mouvement en translation et par blocs

de pixels peut conduire à une mauvaise compensation de mouvement notamment sur les contours des objets. Ces inconvénients détériorent la qualité des images transmises à faible débit. Par conséquent, plusieurs techniques de segmentation ont été développées permettant d'améliorer la compensation de mouvement sur les contours des objets [10][11][12]. D'une manière générale, les techniques de la deuxième génération [13][14] comme les algorithmes division-fusion «split and merge» [15], les transformations de régions [16][17][18], le codage de blocs à taille variable [19], le codage pyramidal [13], ou le codage basé sur le modèle 3D [20] exploitent les avantages des caractéristiques du système visuel en tenant compte des discontinuités sur le bord des objets. Ces techniques sont appliquées avec succès dans le codage des séquences d'images. Néanmoins, elles nécessitent un calcul complexe et la transmission des informations concernant la segmentation ainsi que le codage d'un groupe de pixels. Pour remédier à cet inconvénient, nous proposons une technique de segmentation de bloc qui vise le même objectif que les techniques précédentes mais avec des moyens de calcul plus simples. Elle est basée sur le mouvement des blocs voisins ainsi que sur des modèles décomposables en différents motifs définissant deux parties dans un bloc. Cette technique associée à la méthode rapide proposée permet d'améliorer la compensation de mouvement sur les contours des objets tout en réduisant le nombre de positions de recherche d'environ 4 fois par rapport à la MRE.

On présente ci-dessous un algorithme d'estimation de mouvement avec fenêtre

de recherche angulaire et les résultats associés. Ensuite, on introduit la technique de segmentation proposée. Enfin, on présente les résultats obtenus par la segmentation associée à l'algorithme rapide proposé dans le dernier paragraphe.

Méthode de fenêtre de recherche angulaire

Principe

L'objectif de la compensation de mouvement est de réduire la redondance temporelle entre les images successives d'une séquence donnée. Les méthodes de correspondance de blocs divisent l'image en blocs de même taille et associent le même vecteur de mouvement à tous les pixels du même bloc. Grâce à sa simplicité et sa facilité d'implémentation matérielle la MRE est souvent utilisée. Elle consiste en la recherche de tous les blocs candidats dans une fenêtre de recherche pré-définie. Cependant, elle conduit à un calcul long pour estimer de grandes amplitudes de mouvement. Pour remédier à cet inconvénient, on propose un algorithme plus robuste que les algorithmes rapides et plus rapide que la méthode MRE. Il est basé sur la même hypothèse que celle des méthodes rapides. Il permet de détecter la direction du mouvement en premier et de continuer ensuite la recherche seulement autour de cette direction.

Définissons tout d'abord quelques notations. Soit $\vec{V}_i(x_i, y_i)$ un vecteur de mouvement donné associé au pixel de position spatiale (x_i, y_i) . Soient α_i l'angle que fait \vec{V}_i avec l'axe des abscisses et β un angle donné compris entre 0 et β_{\max} . Soient $\Delta_{\alpha_i-\beta}$ et $\Delta_{\alpha_i+\beta}$ les droites

données par les équations suivantes :

$$\begin{aligned} \Delta_{\alpha_i - \beta} &: y = x \operatorname{tg}(\alpha_i - \beta) \\ \Delta_{\alpha_i + \beta} &: y = x \operatorname{tg}(\alpha_i + \beta) \end{aligned}$$

On définit la région du plan contenant \vec{V}_i et limitée par les droites $\Delta_{\alpha_i - \beta}$ et $\Delta_{\alpha_i + \beta}$ par :

$$\Gamma_{i,\beta} = \left\{ \begin{array}{l} (x, y) \in N \times N \text{ tel que} \\ (y - x \times \operatorname{tg}(\alpha_i - \beta)) \times \\ \text{signe}(y_i - x_i \operatorname{tg}(\alpha_i - \beta)) \geq 0 \\ \text{et} \\ (y - x \times \operatorname{tg}(\alpha_i + \beta)) \times \\ \text{signe}(y_i - x_i \operatorname{tg}(\alpha_i + \beta)) \geq 0 \end{array} \right\}$$

où la fonction signe est définie par :

$$\operatorname{signe}(a) = \begin{cases} 1 & \text{si } a > 0 \\ -1 & \text{si } a < 0 \\ 0 & \text{si } a = 0 \end{cases}$$

En se basant sur l'hypothèse de la cohérence de mouvement, on se limite en première partie à l'estimation de mouvement selon des directions bien définies (figure 1) dans la fenêtre de recherche telle que :

$$\Gamma = \left\{ \begin{array}{l} (x, y) \in N \times N \text{ tel que} \\ (x, y) \in d(i = 1, \dots, 8) \\ \text{et } -p/q \leq x, y \leq p/q \\ \text{où } p, q \in N \end{array} \right\}$$

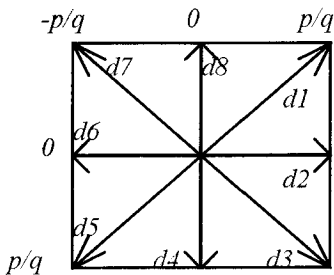


Figure 1. - Estimation de mouvement selon des directions bien définies.

avec :

- d1 : $x = y$ et $x > 0$;
- d2 : $y = 0$ et $x > 0$;
- d3 : $y = -x$ et $x > 0$;
- d4 : $x = 0$ et $y < 0$;
- d5 : $x = y$ et $x < 0$;
- d6 : $y = 0$ et $x < 0$;
- d7 : $y = -x$ et $x < 0$;
- d8 : $x = 0$ et $y > 0$.

Cette étape nous fournit une idée a priori sur la direction du mouvement des blocs.

L'estimation de mouvement est affinée ensuite en utilisant une fenêtre de recherche angulaire autour de la direction du vecteur de mouvement trouvé dans la première étape (figure 2). Cette région est définie par :

$$\Gamma_0 = \{(x, y) \in N \times N \mid -p/q \leq x, y \leq p/q\} \cap \Gamma_{0,\beta}$$

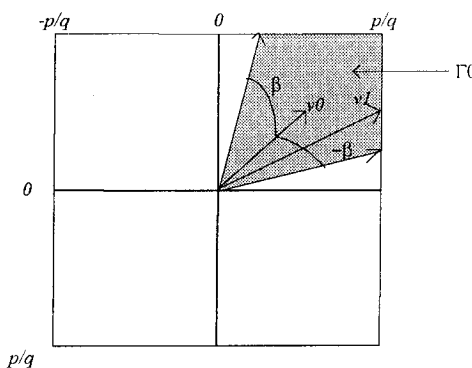


Figure 2. - Estimation de mouvement dans la région $\Gamma_0(q = 2)$

Dans la figure 2, \vec{v}_0 représente le vecteur de mouvement trouvé en première étape et \vec{v}_1 celui trouvé en deuxième étape. En troisième étape on continue la recherche dans la région Γ_1 en utilisant un angle de $\pm\beta$ autour de \vec{v}_1 (figure 3). Cette région est définie

par :

$$\Gamma_1 = \left\{ \begin{array}{l} (x, y) \in N \times N \text{ tel que} \\ -p \leq x \leq p \text{ et} \\ (-p \leq y \leq -p/q \text{ ou} \\ p/q \leq y \leq p) \\ \text{ou} \\ -p \leq y \leq p \text{ et} \\ (-p \leq x \leq -p/q \text{ ou} \\ p/q \leq x \leq p) \end{array} \right\} \cap \Gamma_{1,\beta}$$

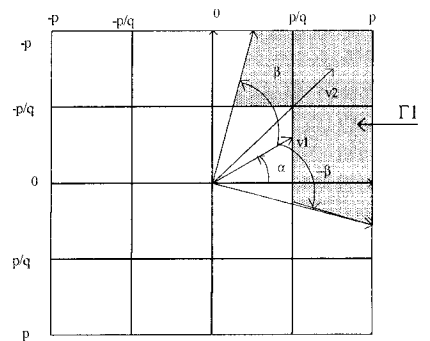


Figure 3. - Estimation de mouvement dans la région $\Gamma_1(q = 2)$.

Evaluation du nombre de positions de recherche

Pour examiner le temps de calcul de l'algorithme proposé, on détermine le nombre de positions de recherche par bloc qui dépend de l'angle β ($0 \leq \beta \leq \pi/4$) et de la taille de la fenêtre de recherche globale $[-p, p]$.

Dans la première étape de cet algorithme, on parcourt les positions de recherche suivant les droites d_i ($i = 1, \dots, 8$). Ceci conduit à $S_1 = 8x(p/2 - 1) + 1$ positions de recherche possibles. Soit $S(\alpha, \beta)$ le nombre de positions de recherche dans la région T_i . Par symétrie de ces régions par rapport aux médianes et aux diagonales, on a $S(\frac{\pi}{2} - \alpha, \beta) = S(\alpha, \beta)$ et $S(\frac{\pi}{2} + \alpha, \beta) = S(\alpha, \beta)$, il suffit donc de calculer $S(\alpha, \beta)$ pour α compris entre 0 et $\pi/4$. Il faut noter que pour α appartenant à cet intervalle, le nombre de positions de recherche $S(\alpha, \beta)$ atteint son maximum pour $\alpha = \pi/4$. Cette

valeur est prise dans le calcul des positions de recherche. Ainsi, en utilisant la région Γ_0 en deuxième étape, on obtient au plus S_2 positions possibles tel que :

$$S_2 = (p/2 + 1)$$

$$[(p/2+1)(1-tg(\pi/4-\beta))-\varepsilon(\beta)] \text{ où } \varepsilon(\beta)$$

correspond aux positions de recherche appartenant à $\Gamma_0 \cap \Gamma$. Enfin, on obtient dans la région Γ_1 , au plus S_3 positions de recherche tel que :

$$S_3 = (3p^2/4 + p)(1 - tg(\pi/4 - \beta))$$

On peut déduire alors le nombre de positions de recherche total NPR tel que $NPR \leq S_1 + S_2 + S_3$.

Résultats et conclusion

Les différentes séquences « test » de l'UER (Union Européenne de Radio diffusion) utilisées (figure 12) sont : *flower*, *mobile & calendar*, *tennis* et *newfoot*. Les images de format initial 720×576 pixels sont divisées en blocs de 16×16 pixels.

La séquence *Mobile & Calendar* contient beaucoup de détails avec un mouvement modéré, ainsi que des structures périodiques.

La séquence *flower* contient une grande variation en profondeur représentant un arbre proche de la caméra et une rangée de maisons s'éloignant davantage avec des fleurs au milieu. Elle correspond essentiellement à un mouvement de la caméra.

L'erreur de prédiction ainsi que les vecteurs de mouvement des images de type *P* sont codés d'une manière similaire à celle de la norme MPEG2, c'est-à-dire avec les mêmes tables de quantification et le même code de Huffman. C'est à partir de ces tables qu'on déduit le nombre de bits réservés au codage de l'erreur de prédiction (BitDFD) ainsi que ceux des vecteurs de mouvement (BitVM).

La fenêtre de recherche est choisie égale à $(\pm 32) \times (\pm 32)$ pixels pour une différence d'images de 3 (images

de type *P*). Le pas de quantification est pris constant. Le critère de correspondance utilisé est l'erreur absolue moyenne (EAM)[2]. La précision de calcul des vecteurs de mouvement est de un demi pixel.

Les simulations comparent la performance de la méthode rapide proposée (MRP) par rapport à la MRE ainsi que les méthodes rapides; 3 étapes (TSS : Three Step Search) et recherche logarithmique (TDL : 2-D Logarithmic) en utilisant les critères suivants : le rapport signal à bruit (SNR (1)), le nombre de positions de recherche par bloc (NPR/bloc) et le nombre de bits total par pixel (bitTOT/pixel) réservé au codage de l'erreur de prédiction ainsi que des vecteurs de mouvement.

$$SNR = 10 \times \log_{10}$$

$$\left(\frac{\sum_n \sum_m (I_t(n, m))^2}{\sum_n \sum_m (I_t(n, m) - \hat{I}_t(n, m))^2} \right) \quad (1)$$

où $I_t(n, m)$ correspond à la luminance du pixel de position spatiale (n, m) .

Le tableau 1 montre que la qualité de la compensation de mouvement de la MRP est satisfaisante. Elle est proche de celle de la MRE tout en réduisant le nombre de positions de recherche de 3 à 7 fois. Par ailleurs, les méthodes rapides ont conduit à une mauvaise compensation de mouvement (figure 4).

L'angle β est pris entre 15° et 45° . Au fur et à mesure que β augmente, l'algorithme devient performant. Cependant, le nombre de positions de recherche augmente aussi en proportion avec β . Ainsi, les résultats expérimentaux montrent que $\beta = 30^\circ$ respecte bien le critère performance/temps de calcul. La figure 4 montre l'évolution du SNR. Ces valeurs sont obtenues par la méthode proposée (MRP $\beta = 45^\circ$) et les méthodes rapides TSS et TDL en référence à la MRE.

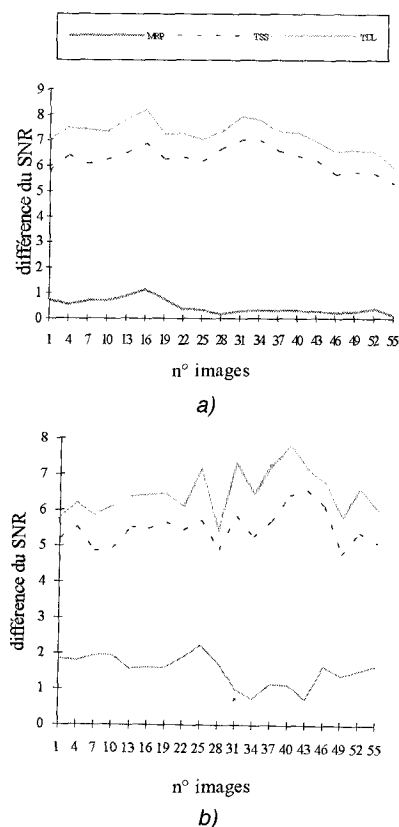


Figure 4. - Courbes de la différence du SNR après compensation par rapport à la MRE, obtenues par la méthode proposée (MRP $\beta = 45^\circ$) ainsi que par les méthodes rapides : TSS et TDL a) Séquence Mobile & Calendar b) Séquence Flower.

Tableau 1. - Valeurs du SNR, du nombre de bits total par pixel (bitTOT/pixel) et du nombre de positions de recherche par bloc (NPR/bloc).

		MRE	TDL	TSS	MRP		
					$\beta = 15^\circ$	$\beta = 30^\circ$	$\beta = 45^\circ$
c	NPR/	4225	37	41	564	901	1160
	a bloc						
e	SNR	24.53	17.3	18.26	23.26	24	24.12
	n [dB]						
d	bitTOT/	0.3399	0.8401	0.7156	0.4166	0.3545	0.3509
	a pixel						
f	SNR	27.25	20.75	21.74	25.37	25.72	26
	l [dB]						
o	bitTOT/	0.4077	0.8106	0.7000	0.4773	0.4606	0.4513
	w pixel						
e	NPR/	4225	37	41	564	901	1160
	r bloc						

En conclusion, la méthode d'estimation de mouvement proposée est plus rapide que la MRE classique tout en réduisant le problème du minimum local. Elle permet un gain de temps de 3 ($\beta = 45^\circ$) à 7 ($\beta = 15^\circ$) fois par rapport à la MRE. Cette méthode sera associée dans la prochaine section à une technique de segmentation afin d'améliorer la compensation de mouvement sur les contours des objets en mouvement.

Technique de segmentation proposée

L'estimation de mouvement uniforme et par bloc de pixels peut conduire à une mauvaise compensation de mouvement notamment sur les contours des objets, car dans ces cas le bloc comporte deux parties dont l'une des deux est en mouvement. D'autre part, un problème de compensation de mouvement de bloc peut surgir lorsqu'un bloc contient deux parties de mouvements différents. Comme il est montré sur la figure 5, la partie foncée du bloc 2 qui est le prolongement d'une partie d'objet contenue dans le bloc 1, doit être compensée à partir du vecteur de mouvement v_1 et non pas v_2 . D'où l'idée d'utiliser le mouvement des blocs voisins et des différents motifs permettant de diviser chaque bloc en 2 parties, afin de corriger la compensation de mouvement autour de ces zones.

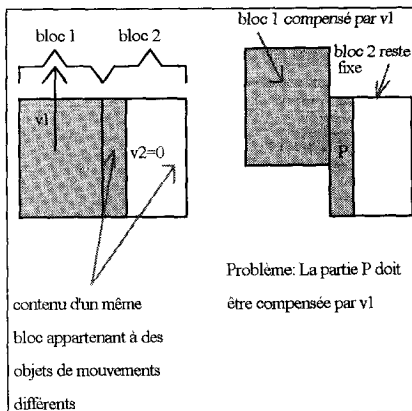


Figure 5. - Exemple de cas où il existe un problème de compensation de mouvement.

Définition du voisinage et choix des motifs

Soit $b_{i,j}$ le bloc de position spatiale (i, j) , l'ensemble de ses blocs voisins est donné par :

$$N_{b_{i,j}} = \{b_{i\pm 1,j}; b_{i,j\pm 1}; b_{i\pm 1,j, j\pm 1}\}$$

Par ailleurs, un motif représente la segmentation d'un bloc 16×16 pixels en deux régions, en utilisant des sous blocs de même taille (8×8 , 4×4 ou 2×2 pixels).

On définit ensuite un modèle A comme étant un ensemble de quelques motifs contenant le même nombre de sous blocs. Ces motifs sont choisis selon les critères suivants :

- les deux régions sont séparées par une seule courbe non fermée.
- les motifs sont caractérisés par deux régions séparées par une ligne verticale, horizontale ou parallèle à la première ou à la deuxième diagonale.
- la symétrie des deux régions d'un bloc permet de ne considérer que la moitié du nombre de motifs possibles.
- on suppose que la taille des blocs est plus petite que celle des objets. Ceci implique que chacune des deux régions du bloc est un prolongement d'une partie d'un bloc voisin, par suite chaque région est adjacente à au moins un bloc voisin.

D'après ces critères, on a choisi trois modèles $A_i (i = 1, 2, 3)$ ne contenant que $6(k-1)$ motifs, qu'on a estimés représentatifs des cas les plus fréquents parmi les $2^{(k \times k)}$ motifs possibles. Les équations décrivant ces motifs sont données par :

$$r_{s,1} : aX + bY < c$$

et $r_{s,2} : aX + bY \geq c$

où $r_{s,1}$ (resp. $r_{s,2}$) définit la région 1 (resp. 2) d'un bloc segmenté selon le motif s d'un modèle donné. Le couple (X, Y) représente les coordonnées d'un sous bloc et les coefficients a, b, c et s caractérisant les motifs sont donnés par le tableau 2 :

Tableau 2. - Paramètres permettant la construction des motifs.

	a	b	c
$1 \leq s \leq (k-1)$	1	0	s
$k \leq s \leq 2(k-1)$	0	1	$s-k+1$
$2k-1 \leq s \leq 4(k-1)$	1	1	$s-2(k-1)$
$4k-3 \leq s \leq 6(k-1)$	1	-1	$s-5(k-1)$

En remplaçant k par 2 dans le tableau ci-dessus, on obtient le premier modèle A1 donné par les 6 motifs suivants (figure 6) :

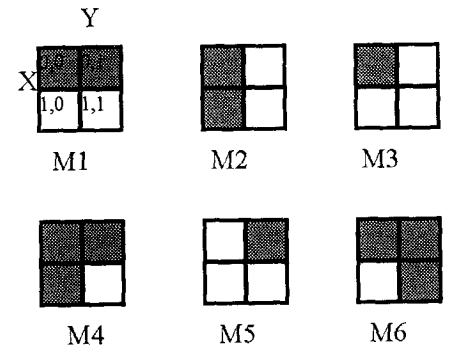


Figure 6. - Modèle A1 contenant 6 motifs.

Par exemple le motif M1 est obtenu pour $a = 1, b = 0$ et $c = 1$ et est décrit par l'équation :

$$r_{1,1} : X < 1 \forall Y \quad \text{et} \quad r_{1,2} : X \geq 1 \forall Y.$$

Les modèles A2 et A3 s'obtiennent pour $k = 4$ et $k = 8$. Il contiennent donc 18 et 48 motifs formés de sous blocs de taille respective 4×4 et 2×2 pixels.

Algorithme de calcul des paramètres de la segmentation

L'estimation de mouvement est faite en premier sur des blocs 16×16 pixels. On partage ensuite chaque bloc en deux régions selon le motif considéré. On associe enfin à chaque région l'un des vecteurs de mouvement des blocs voisins ou celui du bloc lui-même en minimisant la fonction d'erreur (2). L'algorithme est appliqué en deux étapes :

Étape 1 :

Estimation des vecteurs de mouvement des blocs 16×16 pixels.

Étape 2 :

On fixe en premier le motif M_s (par exemple $s = 1$). On partage ensuite le bloc en deux régions $r_{1,1}$ et $r_{1,2}$ selon le motif M_1 (figure 7).

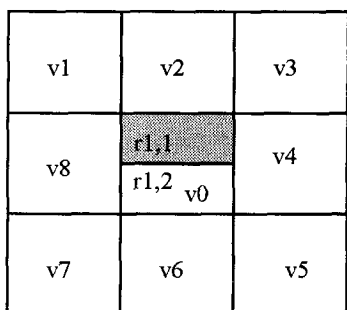


Figure 7. - Division d'un bloc en deux régions selon le motif M_1 .

On associe successivement à chaque région $r_{1,j}$ ($j = 1, 2$) les vecteurs v_i ($i = 0, \dots, 8$). Nous sélectionnons ensuite celui qui minimise la fonction d'erreur suivante :

$$Ed_{s,i,j} = \sum_{(nm) \in r_{s,j}} |DFD(n, m, v_i)|^2 \quad (2)$$

avec :

$$j = 1; s = 1, \dots, 6(k-1)$$

$$\text{et } i = 1, \dots, 8$$

$$DFD(n, m, v_i) = I_t(n, m) - I_{t-1}(n - x_i, m - y_i)$$

$Ed_{s,i,j}$: fonction d'erreur correspondant au vecteur voisin v_i , à la région $r_{s,j}$ et au motif M_s .

On obtient ainsi pour chaque région $r_{1,1}$ (resp. $r_{1,2}$), un vecteur optimal v_1 (resp. v_2) ainsi qu'un minimum des fonctions $Ed_{1,i,1}$ (resp. $Ed_{1,i,2}$) : $\min_{1,1}$ (resp. $\min_{1,2}$).

On note pour $s = 1$:

$$\text{Min}_1 = \min_{1,1} + \min_{1,2}$$

$$v'_{1,1} = v_1; v'_{2,1} = v_2$$

On applique la même procédure à tous les autres motifs M_s ($s = 1, \dots, 6(k-1)$) pour obtenir les minimums Min_s et les

vecteurs $v'_{1,s}$ et $v'_{2,s}$. On déduit enfin le minimum final tel que :

$$\text{Min} = \min_{1 \leq s \leq 6(k-1)} (\text{Min}_s) = \text{Min}_{s_0}$$

On trouve ainsi le motif optimal M_{s_0} ainsi que les deux vecteurs de mouvement correspondant v'_{1,s_0} et v'_{2,s_0} .

L'expression générale permettant d'obtenir les deux vecteurs de mouvement des deux régions et les paramètres a , b et c est la suivante :

$$\text{Min} = \min_{0 \leq s \leq 6(k-1)} \left(\min_{v_i(i=1, \dots, 8)} \sum_{(n,m) \in r_{s,1}} |DFD(n, m, v_i)|^2 + \min_{v_i(i=1, \dots, 8)} \sum_{(n,m) \in r_{s,2}} |DFD(n, m, v_i)|^2 \right)$$

Résultats et conclusion

Les simulations comparent la performance de la méthode rapide proposée (paragraphe 2) avec segmentation par rapport à la méthode de la recherche exhaustive classique, en utilisant deux types de mesure de performance. En premier, la qualité de la compensation de mouvement des images est appréciée par le SNR. En second, on compare le nombre de bits total en tenant compte de l'information supplémentaire réservée à la segmentation et aux vecteurs de mouvement transmis en supplément.

Afin de réduire la quantité d'information supplémentaire réservée à la segmentation (notée par BitSEG dans le tableau 3), on segmente seulement les blocs possédant une erreur de compensation inférieure à 0,9 fois celle sans segmentation.

Les résultats indiqués par le tableau 3 ci-après sont satisfaisants aussi bien en termes d'amélioration du SNR qu'en termes du nombre de bits total utilisés. La méthode rapide proposée associée à la segmentation permet un gain de temps d'environ 4 fois (en tenant compte de la segmentation utilisant le modèle A1)

avec une qualité de compensation de mouvement meilleure que celle de la méthode de recherche exhaustive classique (0.4 à 2.4 dB).

L'évolution du nombre de blocs non segmentés (NBNS) est montrée sur la figure 8, alors que les courbes du SNR et du nombre de bits total par pixel (BitTOT/pixel) incluant la quantité d'information supplémentaire sont montrées respectivement sur les figures 9 et 10 pour les séquences *Mobile & Calendar* et *flower*.

Tableau 3. - Différentes valeurs représentant : le nombre de positions de recherche par bloc (NPR/bloc), le nombre de blocs non segmentés NBNS, le SNR et le BitTOT/pixel. Ces résultats sont obtenus par la méthode proposée (MRP $\beta = 30^\circ$) plus segmentation (A1, A2 et A3) et la MRE en utilisant une fenêtre de recherche de ± 32 pixels pour les séquences a) mobile & calendar b) flower c) tennis d) newfoot.

	Mobile & Calendar			
	MRE	MRP+A1	MRP+A2	MRP+A3
NPR/bloc	4225	987	1203	1703
NBNS	-	66%	56%	52%
SNR [dB]	24.53	26.14	26.68	26.93
BitDFD bits/pixel	0.32104	0.28060	0.27235	0.26779
BitVM bits/pixel	0.01888	0.02793	0.03117	0.03272
BitSEG bits/pixel	0	0.00791	0.01358	0.01688
BitTOT bits/pixel	0.33992	0.31644	0.31710	0.31739

a)

	Flower			
	MRE	MRP+A1	MRP+A2	MRP+A3
NPR/bloc	4225	987	1203	1703
NBNS	-	61%	50%	46%
SNR [dB]	27.25	27.97	28.4	28.58
BitDFD bits/pixel	0.38591	0.37106	0.36480	0.36041
BitVM bits/pixel	0.02380	0.03571	0.03960	0.04012
BitSEG bits/pixel	0	0.00915	0.01543	0.01879
BitTOT bits/pixel	0.40971	0.41592	0.41983	0.41932

b)

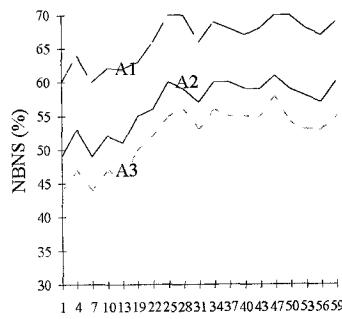
Tennis				
	MRE	MRP+A1	MRP+A2	MRP+A3
NPR/ bloc	4225	987	1203	1703
NBNS	-	65%	54%	50%
SNR [dB]	23.1	23.97	24.51	24.76
BitDFD bits/pixel	0.35847	0.33603	0.33357	0.32997
BitVM bits/pixel	0.02560	0.03720	0.04170	0.04332
BitSEG bits/pixel	0	0.00824	0.01433	0.01753
BitTOT bits/pixel	0.38408	0.38147	0.38960	0.39082

c)

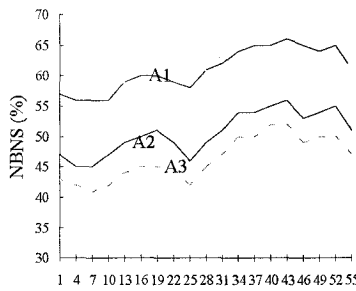
Newfoot				
	MRE	MRP+A1	MRP+A2	MRP+A3
NPR/ bloc	4225	987	1203	1703
NBNS	-	62%	52%	48%
SNR [dB]	24.24	24.66	25.0	23.15
BitDFD bits/pixel	0.41913	0.40377	0.39940	0.39652
BitVM bits/pixel	0.02453	0.03809	0.04343	0.04552
BitSEG bits/pixel	0	0.00888	0.01502	0.01816
BitTOT bits/pixel	0.44337	0.45074	0.45786	0.46020

d)

Pour la séquence *Mobile & Calendar* la segmentation a amélioré la compensation de mouvement obtenue par la méthode de recherche exhaustive (figure 9 a)) ainsi que le nombre de bits total (BitTOT/pixel). L'utilisation du modèle A3 qui est constitué des motifs les plus décomposés a conduit au plus grand nombre de blocs segmentés (48% contre 44% pour A2 et 34% pour A1). En effet, ces motifs ont constitué la meilleure approximation des contours d'objets. Cependant, ces blocs segmentés ont conduit aussi à une quantité d'information supplémentaire (4.3% réservée à la segmentation (BitSEG) et 5.3% pour le codage des vecteurs de mouvement supplémentaires). De ce fait, on a obtenu un nombre de bits total (BitTOT/pixel) pour ce modèle A3 légèrement supérieur à ceux du modèle A2 et A1. Par ailleurs, la segmentation réduit le BitDFD permettant de coder le signal de différence d'environ 12%.

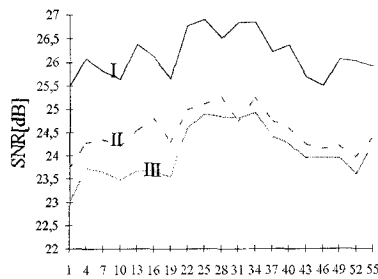


a)

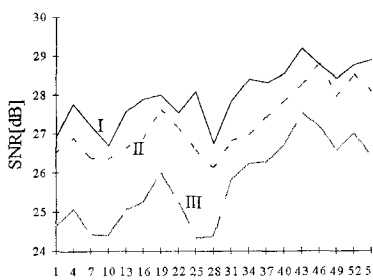


b)

Figure 8. - Courbes représentant le nombre de blocs non segmentés (NBNS) pour les modèles A1, A2 et A3 a) Séquence Mobile & Calendar b) Séquence Flower.



a)

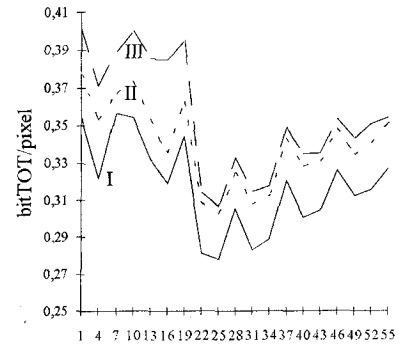


b)

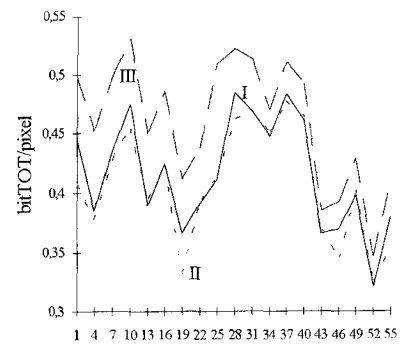
Figure 9. - Courbes du SNR a) Séquence Mobile & Calendar b) Séquence Flower.

Pour la séquence *flower*, une amélioration des valeurs du SNR est réalisée par notre méthode par rapport à la méthode MRE. Les valeurs du nombre de bits total (BitTOT/pixel) sont proches de celles de la méthode exhaustive. Il faut noter que le déplacement de la caméra faisant apparaître des zones couvertes-découvertes conduit à un mouvement non cohérent. La méthode MRE rencontre des difficultés autour de ces zones ainsi que l'algorithme rapide proposé. Bien que la segmentation reste active (39% de blocs segmentés pour le modèle A1, 50% pour A2 et 54% pour A3), le BitDFD n'est réduit que d'environ 4%, ce qui est une valeur juste suffisante pour compenser le BitSEG (2.2%) et les vecteurs de mouvement supplémentaires (2%).

Pour la séquence *newfoot*, la segmentation a amélioré la qualité de compensation de mouvement de 0.4 à 0.9 dB.



a)



b)

Figure 10. - Courbes représentant les valeurs du nombre de bits total par pixel (BitTOT/pixel) a) Séquence Mobile & Calendar b) Séquence Flower.

Alors qu'elle conduit à un nombre de bits total légèrement supérieur par rapport à la méthode MRE. Il faut remarquer que la caractéristique de cette séquence correspond à des déplacements rapides des objets (joueurs ou ballon) ainsi qu'à des changements de plan. Ceci est constaté sur les valeurs du BitVM qui varient rapidement à partir de l'image 49 (figure 11). Cette variation rapide est due à la non corrélation entre les vecteurs de mouvement voisins, et par conséquent une augmentation de la différence entre les vecteurs de mouvement durant le codage. D'autre part, le changement de plan a affecté la valeur du BitDFD entre les images 43 et 46 et pas autant la valeur du BitVM car la plupart des blocs sont codés en intra (70%).

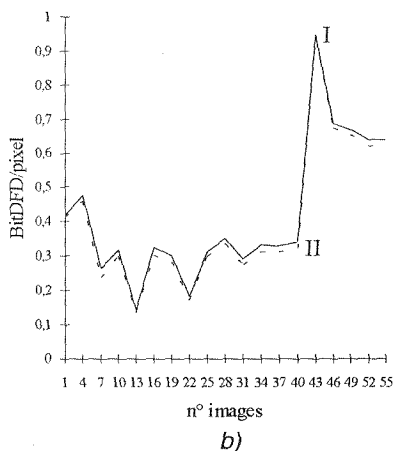
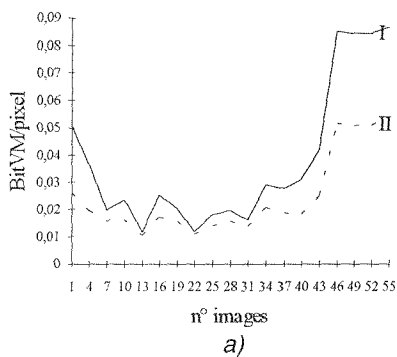


Figure 11. - Courbes représentant : a) le BitVM par pixel b) Le BitDFD par pixel (séquence Newfoot).

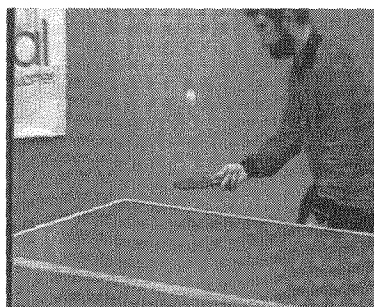
Pour la séquence *tennis*, une amélioration du SNR est constatée (0.9 à 1.6 dB) ainsi qu'un BitTOT légèrement meilleur que celui de la MRE.



a)



b)



c)



d)

Figure 12. - Différentes séquences utilisées : a) calendar, b) flower, c) tennis, d) new-foot.

Il est important de noter que l'algorithme proposé n'améliore pas le SNR d'une manière uniforme à travers les images mais concentre l'amélioration le long des contours des objets en mouvement. Ainsi, le SNR produit par la méthode

proposée a des caractéristiques différentes de celui produit par la MRE. L'algorithme proposé réduit l'effet de blocs (artefacts) dans le signal d'erreur auquel le système visuel est très sensible et qui est difficile à coder par les méthodes de codage standard.

En conclusion, on a proposé un algorithme d'estimation de mouvement pour compenser les défauts d'une part de la méthode exhaustive (temps de calcul long), et d'autre part des méthodes rapides (problème du minimum local). Cet algorithme associé à la technique de segmentation proposée permet la réduction du nombre de positions de recherche d'environ 4 fois tout en améliorant la compensation de mouvement dans un codeur vidéo différentiel par rapport à la méthode exhaustive classique. Il faut noter que la technique proposée comme la méthode exhaustive suppose un mouvement en translation. Cette hypothèse n'est pas toujours respectée. Un modèle de mouvement plus général basé sur des transformations spatiales est en étude.

BIBLIOGRAPHIE

- [1] CCITT RECOMMENDATION H.261, "Video codec for audiovisual services at px64 kbit/s", CCITT, Aug. 1990.
- [2] ISO/IEC JTC1/SC29/WG11 International Organization for standardization. MPEG2 Test Model 1 Technical Report AVC-260, May 1992.
- [3] H.G. Musmann "Advances in picture coding", *Proc. IEEE*, vol 73, n°4, pp. 523-547 April 1985.
- [4] J.R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Trans. Commun.*, Vol. COM-29, pp. 1799-1808, Dec. 1981.
- [5] T. Koga *et al.*, "Motion compensated interframe coding for video conferencing", in *Proc. Nat. Telecommunications Conf.*, New Orleans, LA, Nov. 29-Dec. 3, 1981, pp. G5.3.1-G5.3.5.
- [6] M. Ghanbari "The cross-search-algorithm for motion estimation", *IEEE trans. Comm.*, vol. 38, no. 7, pp. 950-953, July 1990.
- [7] F. Moscheni and J.M. Vesin "A genetic algorithm for motion estimation", 15eme Colloque GRETSI, Juan-Les-Pins, pp. 825-828, Sep. 95.

Informations

- [8] K. H. K. Chow and M. L. Liou "Genetic motion algorithm for video compression", *IEEE Transactions on circuits and systems for video technologie*, vol.3, n°6, pp. 54-70, Dec. 1993.
- [9] Y. L. Chan and W. C. Siu "A new block motion estimation using adaptive pixel decimation", *Proc. ICASSP 1995*, vol. 4, pp. 2257-2260, Detroit Michigan, May 1995.
- [10] M. T. Orchard "Predictive motion-field segmentation for image sequence coding" *IEEE Transactions on circuits and systems for video technologie*, vol 3, n°1, pp. 54-70, Feb. 1993.
- [11] B. Liu, King-Wai Chow, A. Zaccharin "A simple method to segment motion field for video coding" *SPIE*, vol. 1818, pp. 542-549 *Visual Communication and Image processing '92*.
- [12] F. Dufaux and M. Kunt "Multigrid block matching motion estimation with an adaptative local mesh refinement" *SPIE*, vol. 1818, pp.97-108 *Visual Communication and Image processing '92*.
- [13] M. Kunt, A. Ikononopoulos, M. Kocher, "Second generation image-coding techniques" *Proc. IEEE*, Vol.73, n° 4, pp. 549-574 April 1985.
- [14] G. Tziritas and C. Labit "Motion analysis for image sequence coding", Amsterdam; New York : Elsevier, 1994.
- [15] B. Carpentieri and J.A. Storer "Split-merge video displacement estimation", *Proceeding of the IEEE*, vol. 82, n°6, pp. 940-947, June 1994.
- [16] M. Gilge "Region-oriented transform coding (ROTC) of images", *Proc. ICASSP 1990*, vol. 4, pp. 2245-2248 Albuquerque, NM, April 1990.
- [17] H. Nicolas and C. Labit "Region-based motion estimation using deterministic relaxation schemes for image sequence coding" *Proc. ICASSP 1992*, vol. 3, pp. 265-268, California, March 1992.
- [18] J. R. Casas, L. Torres "Coding of Details in Very Low Bit-Rate Video Systems" *IEEE Trans. on Circuits and systems for video technologie*, Vol. 4, n°3, pp. 317-327, June 1994.
- [19] M. H. Chan, Y.B. Yu, A. G. Constantinides "variable size block matching motion compensation with application to video coding" *Proc. IEEE*, vol. 137, PartI, n°4, pp. 2229-2232, August 1990.
- [20] Y. Nakaya and H. Harashima "Motion compensation based on spatial transformations" *IEEE Trans. on Circuits and systems for video technologie*, Vol. 4, n°3, pp. 339-356, June 1994.

Les AUTEURS

Abdallah OUERFELLI, Han VU THIEN, Laboratoire Electronique et Communications, Conservatoire National des Arts et Métiers, 292 rue Saint Martin 75141 Paris cedex 03.

LOUIS - JEAN
avenue d'Embrun, 05003 GAP cedex
Tél. : 04.92.53.17.00
Dépôt légal : 987 — Décembre 1997
Imprimé en France