

Codage vidéo par block matching adaptatif

Abdelhamid Djeffal

Département d'informatique
Université Mohamed Khider
BISKRA, ALGERIE
Abdelhamid_Djeffal@yahoo.fr

Zine Eddine Baarir

Département d'électronique
Université Mohamed Khider
BISKRA, ALGERIE
Info@LESIA.net

RÉSUMÉ. Nous présentons dans cet article une étude des performances des différents algorithmes de Block Matching utilisés dans le codage vidéo tel que Full search, 3SS, GS, 4SS. Un encodeur vidéo est réalisé en but de comparaison de ces algorithmes. Les taux de compression, les PSNR, et les vitesses de codage de ces algorithmes sont comparés en les appliquant sur plusieurs séquences de test. Les résultats obtenus sont exploités pour développer une technique hybride pour l'amélioration de la vitesse d'estimation de mouvement et ainsi la vitesse de codage. Cette technique apprend à partir des codages des images précédentes pour estimer la nature du mouvement d'un bloc pour, d'une part, se rapprocher du bloc de référence optimal et d'une autre part, choisir la méthode de Block Matching adéquate.

ABSTRACT. We present in this paper a performance study of block matching algorithms (BMA) used in video coding such as Full-search, 3SS, GS, 4SS. A video codec is implemented in aim to compare these algorithms. Compression rate, PSNR and coding speed are used in the evaluation, by testing the algorithms on several video sequences. The obtained results are used to develop a fast hybrid technique witch accelerate the motion estimation process, it uses the information gathered from the previous motion estimations to estimate the range of a block motion. The estimated range allows to select the starting search point and the appropriate block matching algorithm for a searched block.

MOTS-CLÉS : Images animées, Compression vidéo, Codage vidéo, Estimation de mouvement, Compensation de mouvement, Block Matching.

KEYWORDS: Animated images, Video compression, Video coding, Motion estimation, Motion compensation, Block Matching.

1. Introduction

L'estimation de mouvement, permettant de coder l'image courante à partir des images I et P, est largement utilisée pour l'élimination de la redondance temporelle entre les images d'une séquence vidéo [2,8,12]. Parmi plusieurs types d'estimation de mouvement, le block matching est adopté par la plupart des standards de codage vidéo tels que H261, H262, H263, H264, H26L, MPEG1, MPEG2, MPEG4,... [1,3,5,11]. Le block matching suppose la subdivision de l'image en blocs non chevauchés de tailles identiques (ou non [9]) et cherche pour chacun le bloc le plus ressemblant dans une image I ou P dite de référence. Généralement, on utilise uniquement la matrice de luminance Y pour la recherche de ces blocs.

Un algorithme de Block Matching commence la recherche à partir d'un point de départ (généralement les mêmes coordonnées du bloc cible) et choisit un ensemble de points candidats autour de ce point selon une distance appelée "pas de recherche". Le pixel, parmi les candidats, minimisant une mesure de distorsion (MDB) est pris comme point de départ de l'étape suivante, et ainsi itère l'algorithme jusqu'à obtenir un bloc satisfaisant ou atteindre les bords de la zone de recherche [6,7]. On trouve plusieurs algorithmes tels que Full Search (FS), 2Dlog search, Orthogonal search, Three Step Search (3SS), Gradient search (GS), Four step search (4SS)...etc.

2. Comparaison des algorithmes de Block Matching

L'amélioration de tels algorithmes passe par leur comparaison pour tirer leurs propriétés, avantages, inconvénients et les types de séquences pour lesquelles ils donnent les meilleurs résultats. Nous avons utilisé pour la comparaison de ces algorithmes des séquences couleur (Figure 1) contenant la plupart des caractéristiques des séquences du monde réel.



Figure 1 Séquences de test utilisées

La comparaison nécessite en premier lieu une plate-forme sur laquelle on peut exécuter ces algorithmes c'est-à-dire un encodeur/décodeur (Codec) vidéo capable de donner des mesures de performance pour chaque algorithme pour différentes séquences. Pour cet objectif, un encodeur est réalisé (Figure 2) en se basant sur les principes Intra et Inter utilisés dans le codage vidéo.

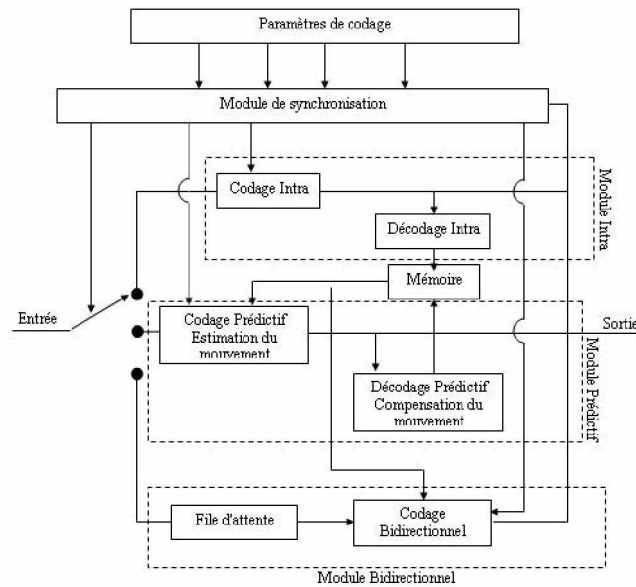


Figure 2. Schéma de l'encodeur utilisé

3. Résultats et discussion

Le tableau 1 résume les résultats obtenus, ces résultats montrent que l'algorithme Full search est très gourmand en temps de calcul, mais donne toujours les meilleurs résultats en taux de compression et en PSNR. Dans le cas des séquences rapides (Carphone) l'algorithme en quatre pas 4SS peut donner des PSNR et des taux de compression similaires et meilleurs que les autres algorithmes en un temps très réduit. Dans le cas des séquences lentes (Miss America) l'algorithme de recherche descendante de gradient GS, est plus avantageux puisque les meilleurs blocs sont toujours très proches du bloc cherché. Dans les séquences contenant les deux types de mouvement rapide et lent (News, Foreman), les résultats sont très proches et la vitesse peut être améliorée en considérant des méthodes de recherche adaptées pour chaque type de bloc.

Séquence \ Algorithme		Algorithme			
		FS	3SS	4SS	GS
Carphone	PSNR	35.29	35.19	35.12	35.27
	TC	94.05	93.99	93.90	94.03
	Blocs Testés	13595825	356794	131040	164685
Miss America	PSNR	34.37	33.78	33.07	34.22
	TC	99.70	99.64	99.63	99.61
	Blocs Testés	73904060	1681772	453448	632960
News	PSNR	34.67	34.45	34.47	34.61
	TC	96.52	96.49	96.46	96.50
	Blocs Testés	13595825	329225	155059	152080
Foreman	PSNR	34.68	34.58	34.65	34.63
	TC	94.01	93.96	93.88	93.96
	Blocs Testés	13595825	400673	193239	308613

Table 1. Comparaison des algorithmes de Block Matching

Dans la littérature, une méthode adaptative est proposée [4], elle utilise un classement des blocs basé sur la valeur de l'erreur entre le blocs cherché et le bloc des mêmes coordonnées de l'image de référence, si cette valeur est inférieure à un minimum, alors le bloc est considéré stationnaire, si elle se situe entre ce minimum et un maximum alors il est considéré lent, sinon il est considéré rapide. Cette méthode n'a pas pu dépasser les limites de 48.11 % pour l'amélioration du nombre de blocs testés par rapport aux méthodes GS et 4SS.

4. Méthode hybride

Dans cette méthode, on classe les blocs au lieu de classer les séquences. Un bloc est classé donc à caractère de mouvement rapide, lent ou stationnaire pour choisir ensuite l'algorithme qui convient pour la recherche du bloc original. Avant d'estimer le vecteur de mouvement d'un bloc, on doit : (1) estimer son caractère de mouvement en se basant sur les informations collectées lors des codages des images précédentes et des blocs voisins précédents dans la même image, (2) choisir la bonne méthode de recherche, (3) choisir la zone de recherche optimale, (4) estimer le point de départ de la recherche pour se rapprocher le plus possible des coordonnées du bloc original. L'estimation du type d'un mouvement d'un bloc (i,j) se base sur une similitude de mouvement, d'une part, avec les blocs: $(i-1,j)$, $(i-1,j-1)$, $(i,j-1)$ de la même image et , d'une autre part, avec le bloc (i,j) dans l'estimation précédente. En utilisant les valeurs des vecteurs de mouvement des blocs précédents déjà estimés, on peut classer donc un bloc comme stationnaire, si la moyenne des vecteurs de mouvement de ses voisins est nulle, ou lent, si cette moyenne ne dépasse pas la zone de 3×3 pixels, ou rapide dans le cas contraire. Ces estimations sont basées sur une table des vecteurs de mouvement trouvés et qui est

mise à jour chaque fois qu'on estime les vecteurs d'une image P. La table des vecteurs de mouvement n'est mise à jour que lors du codage des images P, puisque le codage des images I ne contient aucune estimation, et le codage des images B dispose d'informations postérieures, et par conséquent ses blocs seront éloignés des blocs de l'image P suivante à estimer.

5. Algorithmes considérés

Cas des images P :

L'algorithme suivant permet d'estimer le vecteur de mouvement d'un bloc de coordonnées (X,Y), de taille 16 x 16, d'une image P dans une image précédente d'une distance Dist. L'utilisation de cette distance se justifie par l'utilisation du codage bidirectionnel où la référence ne précède pas immédiatement l'image codée; elles peuvent être séparées par une, deux, trois images voire plus, puisqu'on code des images futures avant des images déjà reçues.

EstimationP (X, Y, Dist)

- 1: Calculer (\bar{X}, \bar{Y}) moyenne des vecteurs du mouvement des blocs : (X-1, Y-1), (X-1, Y), (X,Y-1) et (X,Y) de la table des vecteurs de mouvement
- 2: si $(\bar{X}, \bar{Y}) < (1,1)$ alors Aller à 5
- 3: si $(1,1) \leq (\bar{X}, \bar{Y}) \leq (3,3)$ alors Aller à 6
- 4: si $(\bar{X}, \bar{Y}) > (3,3)$ alors aller à 8
- 5: {Bloc stationnaire} : Calculer la MDB entre le bloc (X,Y) de l'image courante et le bloc (X,Y) de l'image Référence; si $MDB < MinMDB$ alors retourner (0,0)
- 6: {Bloc lent} : Rechercher $(\Delta x, \Delta y)$ par la méthode GS dans une zone de 3x3 pixels en commençant à partir du point $(X * 16 - \bar{X} * Dist, Y * 16 - \bar{Y} * Dist)$.
- 7: Calculer MDB entre le bloc (X, Y) de l'image courante et le bloc (X+ Δx , Y+ Δy) de l'image précédente, si $MDB < MinMDB$ alors retourner $(\Delta x, \Delta y)$
- 8: {Bloc rapide} : Rechercher $(\Delta x, \Delta y)$ par la méthode 4SS dans une zone de 16x16 pixels en commençant à partir du point $(X * 16 - \bar{X} * Dist, Y * 16 - \bar{Y} * Dist)$.
- 9: Retourner $(\Delta x, \Delta y)$.

Cas des images B :

Dans le cas des images B, on dispose d'une référence passée et une référence future, ce qui nous donne plus de confiance sur le type du bloc et nous permet de ne faire aucun approfondissement de la recherche, on s'arrête donc sur la première décision. L'algorithme suivant permet d'estimer un bloc de coordonnées (X,Y) d'une image B

dans une image de référence précédente d'une distance Dist (négative) ou une image suivante d'une distance Dist (positive). Le bloc le plus ressemblant (de l'image précédente ou suivante) est pris comme bloc de référence.

EstimationB (X, Y, Dist)

1: Calculer (\bar{x}, \bar{y}) moyenne des vecteurs du mouvement des blocs (X-1, Y-1), (X-1, Y), (X, Y-1) et (X, Y) de la table des vecteurs du mouvement.

2: si $(\bar{x}, \bar{y}) < (1,1)$ alors retourner (0,0)

3: si $(1,1) \leq (\bar{x}, \bar{y}) \leq (3,3)$ alors Aller à 5

4: si $(\bar{x}, \bar{y}) > (3,3)$ alors aller à 7

5: {Bloc lent} : Rechercher $(\Delta x, \Delta y)$ par la méthode GS dans une zone de 3x3 pixels en commençant du point $(X * 16 + \bar{X} * Dist, Y * 16 + \bar{Y} * Dist)$.

6: Retourner $(\Delta x, \Delta y)$

7: {Bloc rapide} : Rechercher $(\Delta x, \Delta y)$ par la méthode 4SS dans une zone de 16x16 pixels en commençant à partir du point $(X * 16 + \bar{X} * Dist, Y * 16 + \bar{Y} * Dist)$.

8: Retourner $(\Delta x, \Delta y)$.

6. Résultats et discussions

Les résultats de la méthode hybride par rapport aux autres méthodes de block matching sont présentés en terme de PSNR, taux de compression et nombre de blocs testés (temps de calcul) dans le tableau suivant (Table 2).

Séquence	Algorithme	4SS	GS	Recherche hybride	Amélioration	
					Différence	%
Carphone	PSNR	33.63	33.87	33.65	0.02	0.06
	TC	95.06	95.20	94.87	-0.19	-0.20
	Blocs Testés	135383	165106	73922	61461	45.40
Miss America	PSNR	33.38	35.05	33.11	-1.94	-5.53
	TC	99.57	99.56	99.56	0.00	0.00
	Blocs Testés	794797	1112395	168124	945271	84.90
News	PSNR	35.16	35.26	35.16	-0.10	-0.28
	TC	96.45	96.45	96.35	-0.10	-0.10
	Blocs Testés	157152	152080	38877	93403	61.42
Foreman	PSNR	35.24	35.89	35.62	0.38	1.08
	TC	94.31	94.49	94.01	-0.30	-0.32
	Blocs Testés	239514	315127	185885	53429	22.33

Table 2. Résultats du codage par la méthode hybride

La méthode hybride permet d'améliorer le temps de calcul jusqu'à 84.90% (séquence Miss America), sans perte significative en PSNR (1.94 dB) et sans perte en

taux de compression. Dans les autres séquences, on obtient aussi des améliorations très intéressantes en nombre de blocs testés. Le nombre des opérations nécessaires pour la recherche est, dans les cas le plus défavorable, celui de l'algorithme 4SS en cas des mouvements larges, mais dans une séquence lente, le nombre est très réduit. Les opérations de décision relatives au choix de la méthode ou du point de départ sont négligeables en les comparant avec les opérations de comparaison des blocs ou les opérations de leur codage.

7. Conclusion

Dans ce travail, un logiciel de codage vidéo permettant l'étude et la comparaison des algorithmes de Block Matching est réalisé en se basant sur la compression par DCT des images Intra et sur la compression prédictive et bidirectionnelle pour les images Inter. Les algorithmes de Block Matching les plus utilisés sont implémentés tel que Full Search, Descent gradient search, Tree Step search, four step search,... etc. Il s'est avéré que le bon fonctionnement de ces algorithmes est limité à un type précis de séquence, et dans le cas des autres types ils donnent de mauvais résultats: L'algorithme GS convient pour les séquences lentes, tandis que l'algorithme 4SS convient pour les séquences à large mouvement (rapides). Nous avons proposé une méthode hybride consistant en la reconnaissance des types de mouvements des blocs pour choisir l'un des algorithmes précédents pour l'utiliser afin d'optimiser la recherche. Cette reconnaissance est basée sur les informations collectées des mouvements du même bloc dans les images précédentes et de ses voisins dans la même image. Ces informations n'ont pas permis seulement d'estimer le type du mouvement d'un bloc, mais aussi d'estimer le point de départ de la recherche, ce qui a permis de réduire considérablement le temps de calcul. Ce premier pas vers l'apprentissage à partir des images précédentes pour estimer le type de mouvement des blocs dans une séquence animée peut être développé pour profiter des techniques les plus avancées de l'apprentissage tels que les réseaux de neurones, ou même les techniques d'immersion basées sur les pixels ou sur les blocs.

8. Bibliographie

- [1] A.Hamosfakidis, Y.Paker, *A Novel Hexagonal Search Algorithm for Fast Block Matching Motion Estimation*, EURASIP Journal on Applied Signal Processing 2002: 6, 595–600 2002 Hindawi Publishing Corporation
- [2] F.Moschetti, *A Statistical Approach to Motion Estimation*, Thèse PHD, Ecole polytechnique Fédérale de Lausanne 2001.

- [3] G. Coté, L. Winger, *Progrès récents Dans le domaine de la compression vidéo*, IEEE Canadian Review Printemps 2002.
- [4] H. C. Kwan, *Fast Motion Estimation Techniques for Vidéo Compression*, Master of Philosophy Thesis, City University of Hong Kong, July 1998.
- [5] ITU-T SG15, *Video Codec for Audiovisual Service at Px64 Kbits/s*, In ITU-T recommandation H.261 Version 3, Mars 1993.
- [6] E. G. Richardson, Introduction to Image and Video Coding, 2001-2002 at www.vcodex.com
- [7] J. G. Apostolopoulos, *Video Compression: Principles, Practice, and Standards*, Streaming Media Systems Group, HP Labs, Palo Alto, Sept 2005.
- [8] M. Cagnazzo, T. André, M. Antonini, and M. Barlaud, *A model-based motion compensated video coder with JPEG2000 compatibility*, In IEEE Intern. Conf. on Image Processing, pages 2255–2258, Singapore, October 2004.
- [9] R. A. Packwood, M.K.Steliaros and G.R.Martin, *Variable Size of Block Matching Motion Compensation for Object-Based Video Coding*, IPA97, 15-17 July 1997 Conference Publication N° 443 P56.
- [10] S. W. Jang, M. Pomplun, and H. Choi, *Adaptative Robust Estimation of Model Parameters from Motion Vectors*, Proceedings of International Conference on Rought Sets and Current Trends in Computing, Banff, Canada, Vol 2005, December 2001 p 438-441.
- [11] T. Ebrahimi, *Compression vidéo*, Signal Processing Laboratory Swiss Federal Institute of Technology, Lausanne, 2000.
- [12] X. Marichal, *Motion Estimation and Compensation for Very Low Bit Rate Coding*, Thèse PHD, Université Catholique de Louvain, Mai 1998.

9. Biographie

Djeffal Abdelhamid : Magister en informatique option IA et images de l'université de Biskra Algérie 2004, Ingénieur d'état en informatique issu de l'institut national d'informatique à Alger option systèmes informatique en 1997, Maître assistant au département d'informatique à l'université de Biskra depuis Décembre 2004, Membre du laboratoire LESIA et de l'équipe de recherche en traitement des images satellitaires depuis Jan 2005.

Baair Zine Eddine : Docteur-Ingénieur en Electronique option Traitement du Signal de l'Ecole Supérieure d'électricité (SUPELEC), Paris, France 1986, Directeur du laboratoire de recherche LESIA et responsable de l'équipe "Imagerie" au sein du même laboratoire à l'université de Biskra, a dirigé plusieurs projets de recherche à l'université de Biskra et au CDTA à Alger depuis 1988, MACC au département d'électronique à l'université de Biskra.