

Compensation de mouvement sur maillage rectangulaire

Séverine DUBUISSON, Franck DAVOINE

Laboratoire Heudiasyc, U.T.C.,
BP 20529, 60205 COMPIEGNE Cedex
sdubuis@hds.utc.fr, fdavoine@hds.utc.fr

Résumé – Ce papier décrit une méthode de partitionnement d’images en rectangles, ainsi que son utilisation pour la prédiction d’images par compensation de mouvement. Le partitionnement rectangulaire présente l’avantage d’être correctement adapté à la texture de l’image et de générer des blocs de tailles différentes. En prenant en compte ces propriétés, nous testons différentes adaptations des algorithmes de Block-Matching dans le but d’améliorer la qualité de reconstruction des images. Trois méthodes sont décrites, ainsi qu’une optimisation de l’étape d’estimation de mouvement. Enfin, nous testons la robustesse de ces algorithmes pour différentes tailles de périodes de rafraîchissement.

Abstract – This paper presents the partitioning of greyscale images using rectangular partitions, and its use for image prediction by motion compensation. Rectangular partitioning presents the advantage to be well adapted to the image texture and to generate variable size blocks. Taking into account these properties, we test different adaptations of the classical BMA to the rectangular partition, in order to improve the motion compensation performances. Three methods are tested, as well as an improvement of the motion estimation step. Finally, we test the robustness of these algorithms by using different refreshing period sizes.

1 Introduction

Dans le contexte actuel de codage bas-débit, la transmission de séquences vidéo doit se faire le plus rapidement possible. Pour cette raison, il est nécessaire de minimiser le volume d’informations à transmettre au codeur. Une solution communément employée est de regrouper les pixels présentant des caractéristiques communes (à définir) au sein de mêmes régions. Ainsi, lors de l’étape d’estimation de mouvement, plutôt que d’associer un vecteur mouvement à chaque pixel, on en associe un à un groupe de pixels. Une des techniques habituelles est d’abord de diviser l’image courante, issue de la séquence, en blocs réguliers ou non, ensuite, d’effectuer la recherche, pour chaque bloc de la partition, du bloc le plus similaire (selon des critères à définir) dans une image antérieure (le plus souvent, dans l’image immédiatement précédente). Le but est de trouver le meilleur bloc, c’est-à-dire celui minimisant l’erreur quadratique entre ces deux blocs. Ces techniques utilisent des algorithmes dits de Block-Matching.

Il existe différentes façons de diviser une image [4], par exemple à l’aide de blocs triangulaires ou carrés, ces partitions pouvant être ou non adaptées au contenu, ou texture, de l’image. Y. Fischer [2] a proposé un partitionnement rectangulaire adapté à la texture (au contenu) de l’image. En utilisant des algorithmes de Block-Matching (BMA), le mouvement est décrit comme un déplacement (type translation) ou une déformation (type transformation affine ou bilinéaire) des blocs. Une fois que l’on a pu estimer le mouvement, la méthode dite de compensation de mouvement permet de prédire l’image courante en remplaçant les pixels du bloc courant par ceux du bloc “le plus similaire” dans l’image antérieure.

Ce papier explique, dans la première partie, le principe du partitionnement rectangulaire puis décrit, dans la deuxième partie, les différents algorithmes de compensation du mouvement développés, à partir de cette partition, ainsi qu’une

optimisation de l’étape d’estimation de mouvement. Enfin, des résultats expérimentaux sont donnés dans la troisième partie.

2 Le partitionnement rectangulaire

Y. Fischer [2] a développé un partitionnement, appelé partitionnement H-V (Horizontal-Vertical), qui est composé de blocs rectangulaires de tailles variables. Ce partitionnement est dit semi-rigide car, bien que s’adaptant correctement au contenu de l’image (contrairement au quadtree ou au partitionnement carré régulier), les frontières des blocs sont horizontales et verticales (contrairement au partitionnement triangulaire), ce qui lui garde une certaine rigidité. L’idée principale est de faire correspondre les frontières des blocs rectangulaires avec celles de l’image. Cette caractéristique fait que ce partitionnement s’adapte au contenu de l’image, ce qui rend souvent plus facile l’étape d’estimation de mouvement. Une frontière à l’intérieur d’une image est caractérisée, en terme de luminance, par une forte transition de niveaux de gris : trouver ces frontières nous permet de déterminer les endroits où l’image doit être divisée en sous-blocs.

Le principe du partitionnement rectangulaire est récursif : on détermine la moyenne des niveaux de gris sur chaque ligne et colonne de pixels d’un bloc considéré. En calculant la valeur absolue de la différence de la moyenne des niveaux de gris entre deux lignes ou deux colonnes adjacentes - que l’on nomme biais (horizontal ou vertical) -, on met en évidence les fortes transitions de niveaux de gris, c’est-à-dire les frontières de l’image, présentes à l’intérieur du bloc considéré. On peut ensuite diviser le bloc selon la plus forte transition (horizontale ou verticale), puis répéter le processus sur les deux nouveaux sous-blocs engendrés, jusqu’à ce que chaque bloc satisfasse un critère d’homogénéité (cela se traduit par la fixation d’un seuil sur l’écart type entre les niveaux de gris

des pixels du bloc) que l'on aura au préalable défini. Cependant, pour éviter de diviser en deux le bloc trop près de ses frontières, on utilise un facteur de pondération donnant plus d'importance aux frontières de l'image situées vers le centre du bloc. Ainsi, si l'on considère un bloc comme une matrice $M \times N$, que l'on appelle $p_{i,j}$ le pixel intersection de la ligne j et de la colonne i , et que l'on note $\sum_{i=0}^N(p_{i,j})$ la somme des luminances des pixels de la ligne j , on peut définir un coefficient de biais pour chaque ligne (h_j) et chaque colonne (v_i) en utilisant les équations :

$$h_j = \frac{\min(j, M - j - 1)}{M - 1} \left(\sum_{i=0}^N(p_{i,j}) - \sum_{i=0}^N(p_{i,j+1}) \right)$$

$$v_i = \frac{\min(i, N - i - 1)}{N - 1} \left(\sum_{j=0}^M(p_{i,j}) - \sum_{j=0}^M(p_{i+1,j}) \right)$$

Cette manière de diviser une image présente l'avantage d'engendrer une partition flexible qui peut s'adapter de manière correcte au contenu de l'image (voir le cas des Figures 1 et 3), contrairement au partitionnement quadtree, basé sur un processus récursif de division de l'image sans réellement prendre en compte son contenu (on divise en quatre sous-blocs identiques un bloc s'il est considéré comme non homogène), et qui est donc plus rigide. Nous avons simplement ajouté à cette méthode la possibilité de diviser un bloc en quatre sous-blocs s'il est traversé, dans sa diagonale, par une frontière de l'image. Au lieu de garder le plus grand biais vertical ou le plus grand biais horizontal, nous gardons les deux, et effectuons une division en quatre rectangles de tailles différentes le bloc considéré au niveau de ces biais. La figure 1 montre le résultat du partitionnement rectangulaire sur deux images fixes : *Lena* et *Aircraft*, ainsi que sur deux images issues des séquences vidéo *Susie* et *Miss America*.

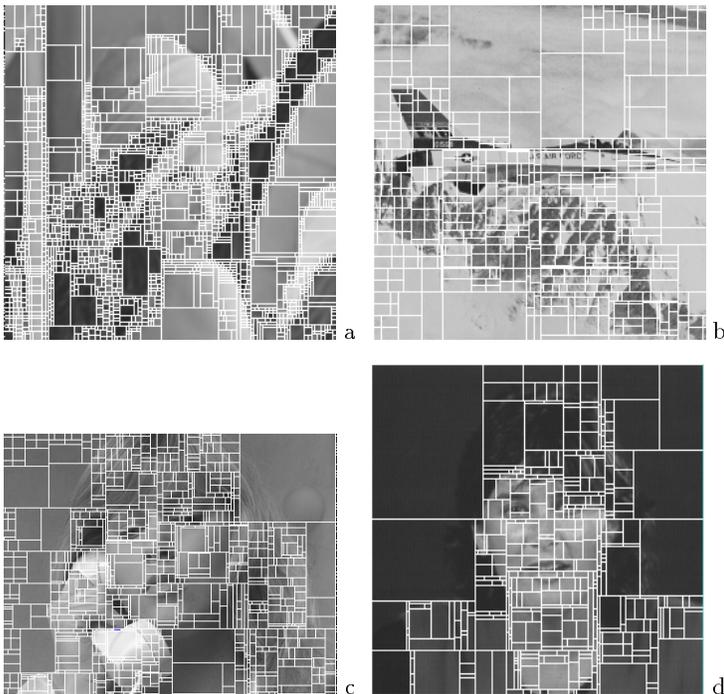


FIG. 1 - Quatre exemples de partitions rectangulaires - a : Image *Lena*. b : Image *Aircraft*. c : Image de la séquence *Susie*. d : Image de la séquence *Miss America*.

3 Estimation de mouvement

De nombreux algorithmes d'estimation de mouvement local sont proposés dans la littérature [8, 7], utilisant des techniques de Gradient, Pel-Recursive ou encore des algorithmes de Block-Matching. Nous avons estimé le mouvement en utilisant plusieurs méthodes, toutes basées sur des algorithmes de Block-Matching, et ce en utilisant des modélisations du mouvement différentes. Les algorithmes de Block-Matching [8] ont pour but la mise en correspondance de deux blocs entre deux images, le principe étant de minimiser une mesure de disparité (l'erreur quadratique moyenne) entre ces deux blocs : tous les pixels à l'intérieur du bloc considéré sont assignés du même vecteur mouvement. Nous expliquons ci-dessous le principe des algorithmes développés :

- La première méthode (BMA-T) est basée sur un algorithme d'estimation de mouvement utilisant un modèle de mouvement dit de translation. On a contraint la recherche à neuf positions autour du bloc considéré, ces positions dépendant de l'intensité du mouvement contenu dans la séquence. Tous les pixels d'un bloc de la partition sont associés au même vecteur mouvement.
- La deuxième méthode (BMA-T+R) estime le mouvement en utilisant un modèle affine à six paramètres [5] : chaque pixel du bloc suit son propre mouvement (qui peut être différent des autres) selon le vecteur déplacement $d(x, y)$ suivant un modèle affine qui peut être exprimé comme suit [1] :

$$d(x, y) = \sum_{i=1}^3 (\varphi_i(x, y) d_i)$$

où $\varphi_i(x, y)$ dépend de la localisation de trois des sommets du bloc rectangulaire, et d_i est le vecteur déplacement correspondant à chacun de ces sommets. Une telle méthode permet de considérer des mouvements plus complexes comme des rotations ou encore des homothéties. Nous nous sommes limités, dans notre cas, à une transformation affine vue comme la composée d'une translation et d'une rotation, chaque paramètre de cette transformation dépendant, encore une fois, du type de séquence étudiée.

- La troisième méthode (BMA-A) s'appuie sur la deuxième méthode en prenant, en plus, en compte la forme géométrique de chaque bloc. Nous avons observé qu'un des inconvénients du partitionnement rectangulaire est de générer parfois des blocs trop allongés : l'estimation du mouvement de ce type de blocs doit se faire d'une manière différente de celle considérée pour un bloc relativement carré. Les paramètres de la transformation affine sont fixés différemment pour ces blocs allongés. Ainsi, en prenant en compte la forme des blocs, nous adaptons l'étape d'estimation de mouvement au partitionnement utilisé, ce qui permet d'optimiser cette première.
- Finalement, nous avons adapté l'estimation de mouvement en modifiant l'aire de recherche autour du bloc considéré, prenant ainsi en compte l'amplitude du mouvement contenu dans la scène. Cette amélioration permet de s'adapter au mouvement contenu dans la séquence, qui peut varier au cours de celle-ci, mais surtout au fait que les objets de l'image ont un mouvement indépendant les uns des autres.

4 Compensation de mouvement

La prédiction par compensation de mouvement est une manière de diminuer la quantité d'informations à transmettre, surtout si le mouvement est estimé à l'aide d'algorithmes de Block-Matching. On associe à chaque bloc de la partition de l'image courante un

vecteur mouvement, dans le cas de la méthode BMA-T, ou plusieurs (définis par des paramètres dépendant de trois des sommets du bloc rectangulaire ainsi que des coordonnées du pixel dont on cherche la transformée) pour les deux autres méthodes utilisant un modèle de mouvement affine (BMA-T+R et BMA-A - voir la section 3). La compensation de mouvement rétrograde consiste à prédire l'image courante en utilisant les pixels d'une image antérieure, le plus souvent l'image immédiatement précédente. Ainsi, on prédit l'intensité d'un pixel de l'image courante en retrouvant sa position dans l'image précédente (cela se fait pendant l'étape d'estimation de mouvement). Il existe donc entre le pixel prédit et le pixel de l'image précédente une relation de transformation (affine, bilinéaire ou autre) dont on a déterminé les paramètres. La position d'un pixel de l'image prédite correspond donc à la position arrondie de la transformation de son pixel correspondant dans l'image précédente. Le Figure 2 illustre le principe dans le cas d'une transformation simple de translation : on a déterminé le bloc le plus similaire au bloc courant dans l'image précédente, puis associé un vecteur mouvement entre ces deux blocs. Le contenu du bloc recherché est ensuite transvasé dans celui du bloc correspondant.

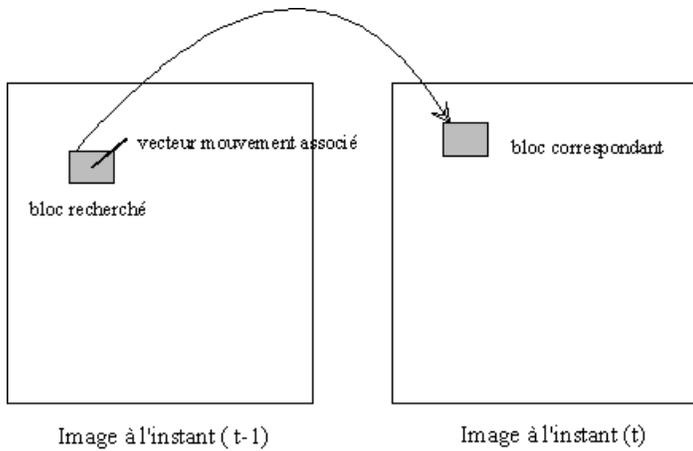


FIG. 2 - Illustration du principe de la compensation de mouvement rétrograde

Afin de déterminer la qualité de prédiction de l'image courante, on calcule le PSNR entre l'image réelle et l'image prédite par la formule suivante :

$$PSNR_{ab} = 10 \log_{10} \frac{255^2 N}{\sum_{i=1}^N (I(p_i) - \tilde{I}(p_i))^2}$$

où N est le nombre total de pixels contenus dans l'image, $\tilde{I}(p_i)$ l'intensité prédite et $I(p_i)$ l'intensité réelle.

5 Résultats

5.1 Etude comparative des différents algorithmes

Nous avons d'abord testé nos algorithmes sur quelques séquences vidéo comme *Miss America* et *Mother and daughter*. Pour la séquence *Miss America*, le partitionnement engendré est composé en moyenne de 450 blocs. La Figure 3 montre les différentes étapes de nos algorithmes, testés sur cette séquence : tout d'abord, on partitionne l'image courante avec des blocs rectangulaires (voir la section 1), puis on estime le mouvement, ceci en utilisant différents algorithmes (voir la section 2) et enfin nous effectuons la prédiction par compensation de mouvement. Nous essayons, dans les algorithmes utilisant un modèle affine, trois angles différents de rotation (+10, 0 et -10 degrés), ainsi que différents pas de translation.

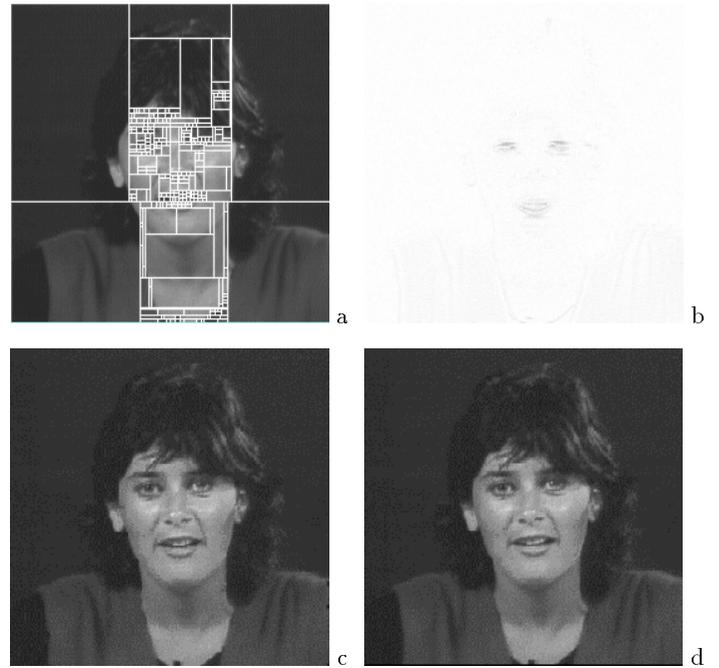


FIG. 3 - a: Une image partitionnée : 450 rectangles. b: Image des différences (image courante - image précédente). c: Image prédite. d: Image courante.

Nous avons aussi effectué des tests comparatifs entre les trois méthodes BMA-T, BMA-T+R et BMA-A sur plusieurs séquences vidéo. La Figure 4 montre la comparaison des PSNR pour les trois techniques testées sur la séquence *Miss America*. On peut voir sur ces figures que la méthode BMA-A donne de meilleurs résultats, en terme de qualité de prédiction, permettant parfois d'augmenter le PSNR de 1.5 dB.

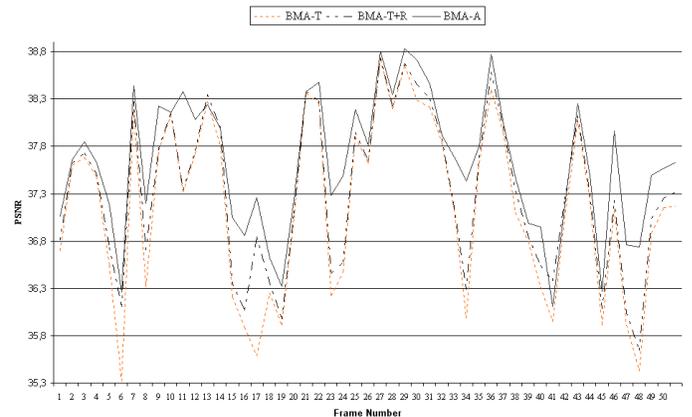


FIG. 4 - Comparaison entre les différents algorithmes pour la séquence *Miss America*.

Jusqu'alors, nous faisons une recherche assez grossière lors de l'étape d'estimation de mouvement. Nous avons affiné cette étape de manière à :

- Considérer que l'amplitude du mouvement change tout au long de la séquence
- Prendre en compte le fait que les objets d'une même scène peuvent avoir un mouvement d'amplitude différente les uns des autres

La Figure 6 montre l'amélioration de la qualité de reconstruction des images de la séquence *Miss America* en fonction de la taille et du nombre de zones de recherche (1, 3 ou 5 zones).

5.2 Compensation de mouvement validée sur un groupe d'images

Il est important de valider nos résultats dans un contexte de codage. La norme MPEG-2 [6] impose un débit qui oblige à minimiser le plus possible le volume d'informations à transmettre au codeur. L'idéal serait de prédire une nouvelle image en utilisant la prédiction de l'image précédente, mais cela propagerait, au fur et à mesure, une erreur trop importante, surtout dans notre cas où l'on ne prend pas en compte l'erreur de prédiction de l'image précédente. On peut cependant utiliser l'image prédite comme image référence (plutôt que l'image réelle) pendant des périodes de longueur fixée (c'est ce qui se fait en codage type MPEG). Nous avons voulu tester ceci sur nos algorithmes, et vérifier que l'erreur qui se propage n'est pas trop importante. La figure 5 montre les comparaisons entre les résultats obtenus pour le partitionnement rectangulaire (en moyenne, 300 blocs) et le partitionnement carré régulier (blocs de 16 pixels sur 16 pixels, soit 256 blocs), ceci pour des périodes de rafraîchissement de 3 et 5 images, sur la séquence *Miss America*. Nous modélisons, ici, le mouvement par une simple translation qui s'étend de 8 pixels de chaque côté du bloc (soit 16 pixels d'amplitude totale horizontale et verticale).

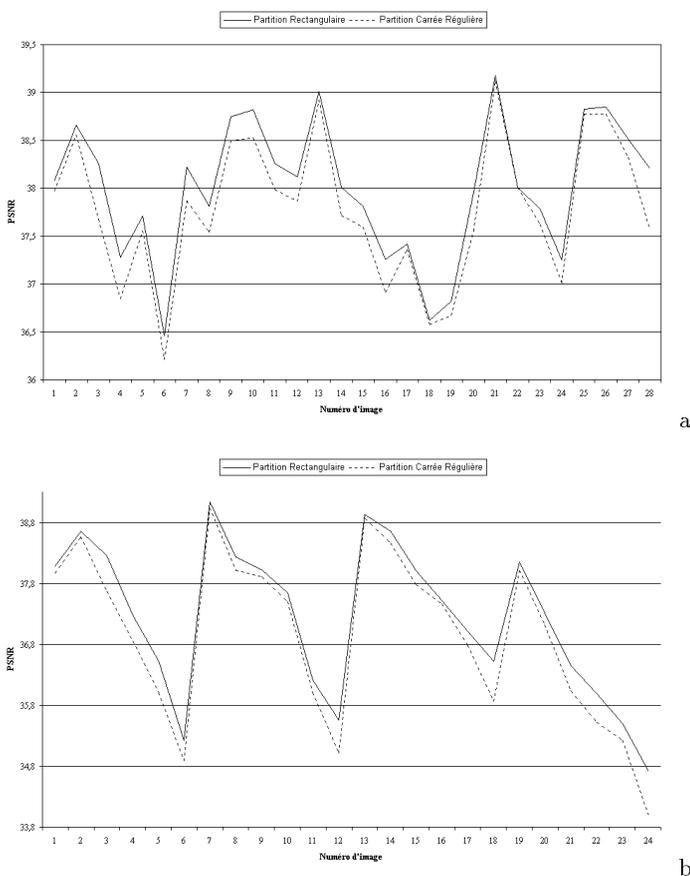


FIG. 5 - Comparaison entre le partitionnement rectangulaire et le partitionnement carré régulier pour différentes périodes de rafraîchissement, séquence *Miss America* - a : Rafraîchissement toutes les 3 images. b : Rafraîchissement toutes les 5 images.

Sans coder l'erreur, les qualités de reconstruction de l'image courante en utilisant le partitionnement rectangulaire restent, et ce même pour des taux de rafraîchissement assez faibles, supérieures à celles obtenues avec un partitionnement carré régulier.

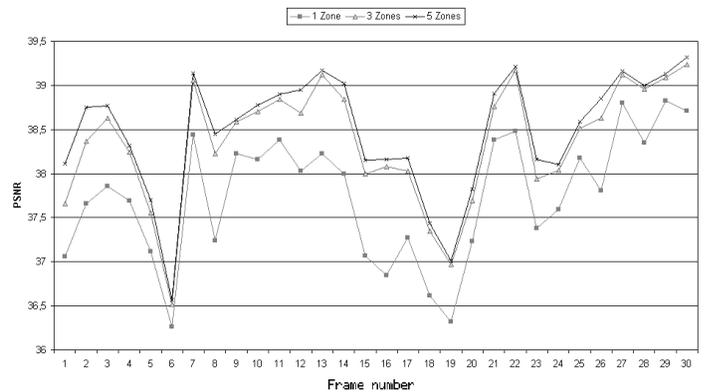


FIG. 6 - Comparaison, pour la séquence *Miss America* entre le BMA-A "classique" (1 seule zone de recherche) et le BMA-A "amélioré" (3 et 5 zones de recherche).

6 Conclusion et perspectives

Ce papier présente un partitionnement rectangulaire efficace qui peut être utilisé à la fois pour le traitement d'images fixes et pour celui de séquences vidéo. Nous avons aussi montré qu'il peut être adapté aux algorithmes courants d'estimation de mouvement. De plus, nous avons mis en évidence que l'étape d'estimation de mouvement, en utilisant un modèle affine, donne de meilleurs résultats si l'on tient compte des caractéristiques géométriques des blocs : l'augmentation de la valeur PSNR, en comparaison avec les méthodes classiques prouve que cette étape a été grandement améliorée par notre méthode. Cependant, on ne peut ignorer le fait que le partitionnement rectangulaire étant adapté au contenu de l'image, il est nécessaire de le coder, contrairement au partitionnement carré régulier ou au quadtree (dont le codage est très simple et peu coûteux).

Nous comptons, dans une prochaine étape, étendre nos recherches vers une approche orientée maillage actif [3], où la prédiction par compensation de mouvement ne se fera plus à l'aide d'algorithmes de Block-Matching, mais par déformation de maillage : on estime la position de chaque sommet à l'aide d'une modélisation bilinéaire du mouvement, ce qui déforme les partitions. Nous envisageons également de travailler sur le codage orienté objet [9], où l'on ne partitionnerait plus toute la scène, mais seulement les objets en mouvement dans celle-ci.

Références

- [1] M. Dudon, O. Avaro, and C. Roux. Triangular active mesh for motion estimation. *Signal processing: Image Communication*, 10:21-41, 1997.
- [2] Y. Fisher, editor. *Fractal Image Compression: Theory and Application to Digital Images*. Springer Verlag, New York, 1995.
- [3] P. Lechat, N. Laurent, and H. Sanson. Représentation d'images et estimation de mouvement basées maillage. application à un codeur tout-maillage. In *Actes des journées CORESA*, 1998.
- [4] G. Marquant, H. Le Floch, and C. Labit. Génération et suivi de maillages adaptatifs: un état de l'art et quelques résultats. In *Actes des journées CORESA*, 1998.
- [5] Y. Nakaya and H. Harashima. Motion compensation based on spatial transformations. *IEEE: Transactions on Circuits and Systems for Video Technology*, 4(3):339-356, jun 1994.
- [6] R. Schafer and T. Sikora. Digital video coding standards and their role in video communications. *Proceedings of the IEEE*, 83(6):907-924, jun 1995.
- [7] A. Sharaf and F. Marvasti. Motion compensation using spatial transformations with forward mapping. *Signal processing: Image Communication*, 14:209-227, 1999.
- [8] A. Murat Tekalp. *Digital Video Processing*. Engelwood Cliffs, NJ:Prentice-Hall, 1995.
- [9] A. Murat Tekalp, P. Van Beek, C. Toklu, and B. Gunsel. Two-dimensional mesh-based visual object representation for iterative synthetic/natural digital video. 86(6):1029-1051, jun 1998.