

DIIC - INC, 3<sup>e</sup> année

—

Module COMV

## TP 4 - compensation de mouvement par *block matching*

Gaël Sourimant, Luce Morin

13 novembre 2006

### 1 Introduction

Dans les schémas de type MPEG ou H26x, la redondance statistique qui correspond à la corrélation temporelle d'un signal vidéo est exploitée. Ces méthodes s'inscrivent dans la famille de codage dite *inter-images*. Les séquences d'image considérées sont telles que le fond peut avoir un mouvement apparent (en raison, par exemple, du mouvement de la caméra) et dans lequel peuvent se trouver des éléments mobiles (visage, véhicule, etc). Dans le TP2, la corrélation spatiale des intensités lumineuses était exploitée au moyen d'un codage DPCM.

Le codage que vous allez considérer dans ce TP présente de ce point de vue des analogies sur le principe. En effet, vous allez coder l'image d'erreur qui correspond à la différence entre deux images successives. Sous l'hypothèse que l'illumination de la scène est constante, les différences entre les images sont alors essentiellement dues aux mouvements dans la vidéo. Mieux : sous l'hypothèse de mouvements apparents 2D d'amplitude raisonnable, il est même possible de calculer une estimation de ces mouvements d'une image sur l'autre, puis de s'en servir comme d'une prédiction temporelle. Vous transmettez alors deux éléments pour reconstruire l'image  $I_{t+1}$  à partir de l'image  $I_t$  :

- la prédiction temporelle du mouvement ( $\rightsquigarrow$  compensation de mouvement, section 2.1),
- l'erreur commise ou signal d'innovation ( $\rightsquigarrow$  codage temporel différentiel, section 2.2).

C'est le moindre coût du signal d'innovation quantifié qui, comme dans le schéma DPCM, permettra le gain en compression.

Les méthodes de mise en correspondance utilisée dans ce TP exploitent directement le signal d'intensité. Elles reposent sur une mesure de similarité entre images successives. En général, ces mesures ont un support de quelques pixels, sous l'hypothèse d'un mouvement localement constant (une translation), afin de rendre l'estimation du mouvement plus fiable, et surtout plus rapide. Vous prendrez pour support de l'estimation du mouvement de translation des carrés de 8 ou 16 pixels de côté (*block...*). C'est à dire que vous vous attacherez à trouver dans l'image  $I_{t+1}$  le bloc le plus similaire dans l'image  $I_t$  (...*matching*).

### 2 Considérations pratiques et principe

Vous travaillerez sur les images rp10.pgm, rp11.pgm et rp20.pgm qui se situent dans le répertoire partagé /share/diic3/inc\_tcomv/images. Remarquez que ces images appartiennent à une même séquence vidéo. Remarquez également que les images rp10.pgm et rp11.pgm sont temporellement consécutives, alors que l'image rp20.pgm est située plus tard dans le flux vidéo.

Comme pour les TP2 et TP3, vous rendrez un compte-rendu comprenant une description et une justification succincte des algorithmes et vos réponses aux questions posées. N'oubliez pas d'insérer en annexe de celui-ci le code source du programme assorti des commentaires idoines.

## 2.1 Compensation de mouvement

Soient deux images en niveaux de gris  $I_t$  et  $I_{t+1}$ . Ces images seront représentées par des tableaux à 2 dimensions (imat). Il s'agit de reconstruire  $I_{t+1}$  à partir de  $I_t$ . Pour cela, l'image  $I_{t+1}$  est découpée en blocs carrés. Pour chacun de ces blocs  $B$  de  $I_{t+1}$ , il s'agit de trouver un bloc dans  $I_t$  tel que l'écart entre les niveaux de gris des 2 blocs soit minimal au sens d'une norme à choisir (voir ci-dessous).

### Modèle de mouvement

Les mouvements réels peuvent *a priori* être de toute nature (la mise en correspondance d'un bloc entre deux images successives devrait idéalement pouvoir prendre en compte des rotations, divergences, etc. en plus de la simple translation.) Des modèles de mouvement complexes peuvent décrire finement le mouvement des blocs, mais ils comportent de nombreux paramètres dont l'estimation précise est rendue difficile sur une taille de bloc petite. En outre, leur coût de calcul devient rapidement prohibitif.

Pour toutes ces raisons, vous vous limiterez donc à un modèle de mouvement translationnel. De plus, pour des raisons de simplicité, vous ne considérez que des translations d'un nombre entier de pixels.

### Critères de similarité

Différents critères peuvent être utilisés pour mesurer la similarité entre deux blocs potentiellement mis en correspondance. Des critères complexes peuvent permettre de s'affranchir des variations temporelles de l'éclairage, mais pénalisent le temps de calcul. Vous vous limiterez ici à des critères de similarité assez simple : si  $p$  désigne un pixel d'un bloc  $B$  et  $T$  un vecteur de translation, vous définirez l'erreur de reconstruction  $E_B(T)$  par une des mesures suivantes.

- Somme des valeurs absolues (SVA, norme  $L_1$ ) :  $E_B^{SVA}(T) = \sum_{p \in B} |I_t(p+T) - I_{t+1}(p)|$
- Erreur quadratique (EQ, norme  $L_2$ ) :  $E_B^{EQ}(T) = \sum_{p \in B} (I_t(p+T) - I_{t+1}(p))^2$
- Erreur maximale (EM, norme  $L_\infty$ ) :  $E_B^{EM}(T) = \max_{p \in B} |I_t(p+T) - I_{t+1}(p)|$

### Fenêtre de recherche

Pour estimer la translation associée à chaque bloc  $B$ , il faut fixer la taille  $L$  de la fenêtre de recherche. En général, aucune information *a priori* n'est disponible pour estimer le mouvement : la fenêtre de recherche sera prise centrée sur le bloc dans  $I_{t+1}$ , et la taille  $L$  sera choisie arbitrairement en fonction de l'amplitude maximale des mouvements attendus dans la scène et des limites imposées par les dimensions de l'image elle-même. Le mouvement retenu  $\hat{T}$  est alors

$$\hat{T} = \arg \min_{|T| \leq L} E_B(T).$$

### Méthode de recherche

Il existe plusieurs manières de chercher le vecteur de mouvement d'un bloc : recherche exhaustive, recherche logarithmique, recherche multigrille, recherche à 3 pas, etc. Par souci de simplicité, on se contentera dans ce TP d'implémenter la méthode exhaustive, où toutes les positions possibles des blocs à mettre en correspondance depuis  $I_{t+1}$  dans  $I_t$  sont explorées, dans une fenêtre de recherche donnée.

Le principe est illustré sur la figure 1 : pour le bloc courant  $B$  de  $I_{t+1}$ , on se positionne à la position correspondante dans  $I_t$ , et on recherche le vecteur de mouvement  $\hat{T}$  qui va minimiser l'erreur entre le bloc  $B$  à la position  $p$  dans  $I_{t+1}$  et le bloc  $B'$  à la position  $p + T$  dans  $I_t$ .

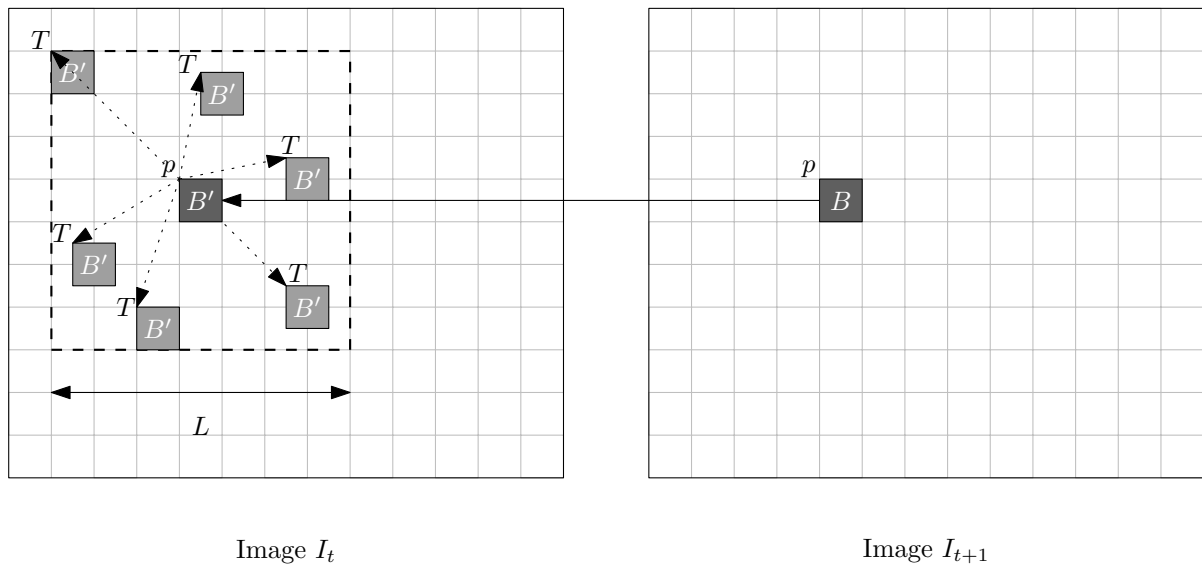


FIG. 1 – Recherche du meilleur vecteur de mouvement

## 2.2 Codage temporel différentiel

Pour compenser les erreurs commises par la compensation de mouvement, l'idée est trivialement de ne coder non pas le bloc courant de l'image  $I_{t+1}$  ou son correspondant dans l'image  $I_t$ , mais plutôt la différence entre ces deux blocs. C'est la quantification de cette erreur qui va permettre de compresser le signal.

## 2.3 Reconstruction

La partie décodage (i.e. reconstruction) est relativement simple. Les informations transmises sur le canal pour l'image  $I_{t+1}$  sont les vecteurs de mouvement correspondant aux déplacements de blocs trouvés, ainsi que l'image de différence entre les blocs de  $I_{t+1}$  et leurs correspondants dans  $I_t$ . On suppose que le décodeur connaît l'image  $I_t$  de référence; il va décoder l'image  $I_{t+1}$  en utilisant cette image de référence conjointement aux vecteurs de mouvements ainsi qu'à l'image de différence qu'il reçoit pour cette image  $I_{t+1}$ .

## 3 Programmation

Un squelette de programme vous est fourni dans `/share/diic3/inc_tpcomv/tp4`.

**Question 1.** *Écrivez un programme qui prend en entrée deux images, la taille des blocs et la taille de la fenêtre de recherche, qui effectue une compensation de mouvement en employant la recherche exhaustive et qui renvoie en sortie les informations suivantes :*

- la seconde image reconstruite à partir de la première;
- des vecteurs de mouvement (sous une forme à définir);
- l'image de différence.

La mise au point de ce programme devrait suivre les étapes ci-dessous.

1. Écrivez une fonction qui calcule l'erreur de reconstruction d'un bloc donné pour une translation donnée. Un bloc est défini par les coordonnées de son premier pixel (en haut à gauche). Pensez à trouver une solution pour traiter le point suivant : un bloc déplacé peut sortir de l'image.
2. Écrivez une fonction qui effectue la recherche du meilleur mouvement pour un bloc donné.
3. Écrivez une fonction qui, à partir de 2 images, reconstruit la seconde à partir de la première, des vecteurs de mouvement, et de l'image d'erreur. Remarquez que la taille d'une image n'est pas nécessairement un multiple de la taille du bloc utilisée. On ne cherchera pas dans le cadre de ce tp à résoudre ce problème, et les images fournies auront des dimensions multiples de tailles de bloc que vous allez utiliser.

**Question 2.** Pour chaque couple d'images et pour les différentes valeurs des paramètres d'entrée,
 

- calculez le PSNR entre l'image originale/reconstruite (à programmer, cf. tp1, tp2 et tp3),
- calculez l'entropie de l'image d'erreur (à programmer, cf. tp2),
- calculez la norme du plus grand vecteur de mouvement (à programmer),
- notez le temps de calcul (cf. le squelette du programme),
- visualisez l'image reconstruite et l'image d'erreur.

 Vous pourrez remplir pour cette question un tableau sur le modèle de celui donné en annexe.

## 4 Questions

- Question 3.** Quelles valeurs des paramètres vous semblent donner le meilleur compromis qualité de reconstruction/temps de calcul ?
- Question 4.** Au vu de vos résultats, quels sont les inconvénients d'une technique de type block matching ?
- Question 5.** Que se passe-t-il si un objet se déplace de moins d'un pixel ?
- Question 6.** Que se passe-t-il si un objet est animé d'un mouvement de rotation ?
- Question 7.** Que suggérez-vous pour remédier à ces problèmes ?

---

⌘ Fin du TP ⌘

## Annexe

### Exemple type de tableau récapitulatifs pour les résultats

Les tailles de blocs  $B$  sont des multiples de deux, mais les tailles de la fenêtre de recherche sont des entiers quelconques et ne sont données ici qu'à titre indicatif. A vous de trouver des paramètres qui vous semblent intéressants à interpréter.

•  $rp10 \rightarrow rp11$  :

tailleB / tailleL :	8 / 10	8 / 50	8 / 100	16 / 10	...
PSNR	...	...	...	...	...
Entropie	...	...	...	...	...
Norme + gd vecteur	...	...	...	...	...
Temps de calcul	...	...	...	...	...

TAB. 1 – Récapitulatif des résultats obtenus avec différents paramètres