

## Partie II : Compression MPEG

### I- OBJECTIF du TP :

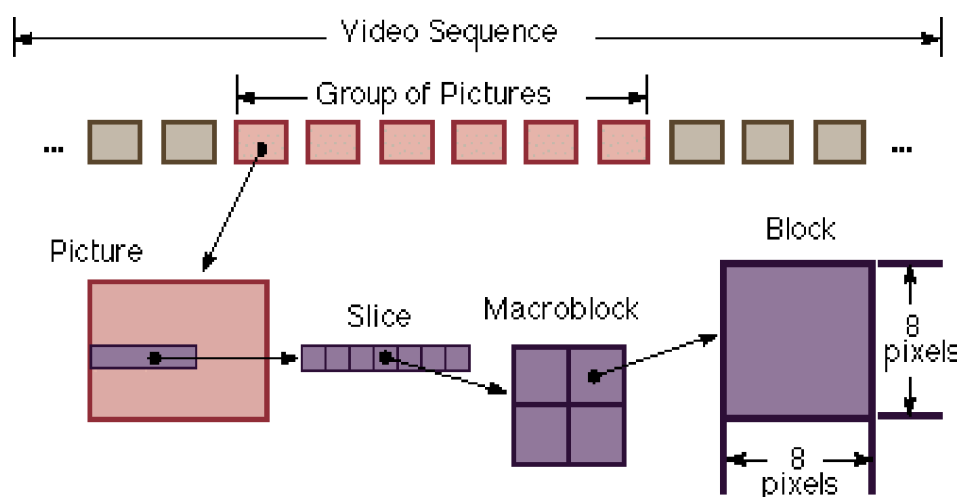
- Initiation à la compression vidéo.
- Implémenter sous Matlab une compression MPEG

### II- RAPPELS THEORIQUES :

#### II.1 La compression MPEG (*Moving Picture Expert Group*)

Le format MPEG consiste à stocker une succession d'images encodées en JPEG. Les images sont mises l'une après l'autre dans le fichier. Comme la compression JPEG n'examine pas les corrélations temporelles, on n'arrive pas à un bon taux de compression global. Les fichiers MPEG sont en général de très grande taille.

**MPEG-2** est la norme de seconde génération (1994) qui fait suite à MPEG-1. MPEG-2 définit les aspects compression de l'image et du son (vidéo) pour les applications liées à la télévision numérique. Généralement, ces techniques de compression se composent de trois parties : une partie vidéo, une partie audio, et une partie système qui gère l'intégration des deux premières. Dans le flux vidéo, les données sont hiérarchisées selon une manière bien précise comme montre la figure suivante.



*Fig.1. Constitution d'une séquence vidéo MPEG*

Dans une séquence vidéo, il existe deux sortes de redondances : la redondance spatiale et la redondance temporelle.

- ❖ La **redondance spatiale** est celle dans chaque image prise indépendamment des autres. On peut diminuer cette redondance en codant chaque image séparément en JPEG.
- ❖ La **redondance temporelle**. Le but est de ne stocker que ce qui est modifié lors du passage d'une image à une autre.

Il existe trois types d'images : les images I (images intra-codées) et les images P (images prédictives). Les **images I** sont des images complètes codées en JPEG. Les **images P (Prédictives)** sont Codées par rapport à l'image I ou P qui la précède, grâce aux techniques de prédiction avec compensation de mouvement. Les **images B (Bidirectionnelles)** Codées par interpolation entre les images I ou P qui les encadrent.

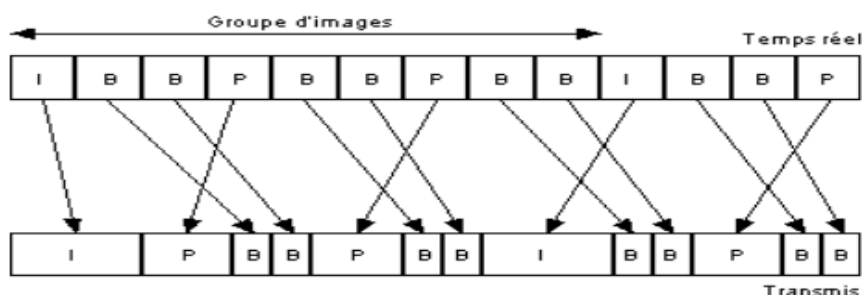
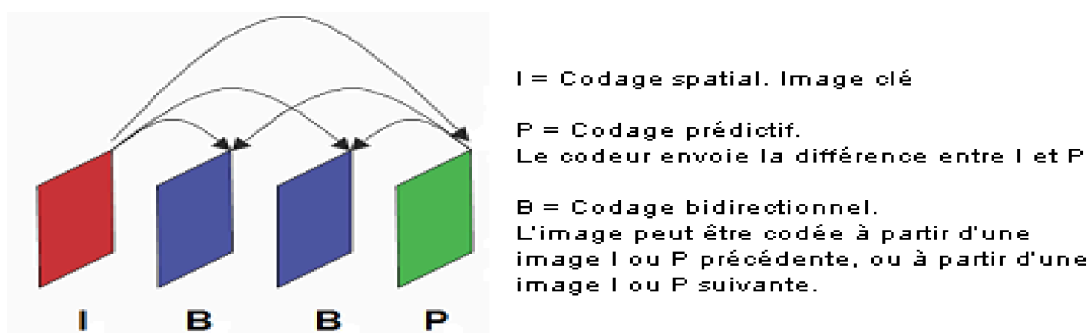
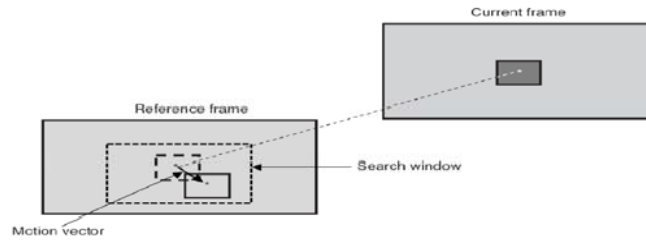


Fig.2. Composition d'un flux vidéo

## II.2 Estimation de Mouvement

L'estimation de mouvement consiste à trouver un vecteur assurant la correspondance entre une zone de départ sur une image et une zone d'arrivée sur la suivante. Les zones de recherches sont des carrés de 16 pixels de côté appelés 'Macroblocs'. On déplace le bloc de départ dans une fenêtre de recherche de manière à trouver le macrobloc le plus ressemblant (*Block matching*) dans l'image de référence ce qui engendre un vecteur de mouvement possédant des composantes dans les directions horizontales et verticales. Comme l'opération du *Block matching* est coûteuse et le mouvement est généralement petit entre les trames adjacentes, la recherche de similitude entre blocs est effectuée dans une fenêtre de recherche de taille limitée. La Figure ci-dessous illustre cette technique.



**Fig.3. Opération du Block matching**

Plusieurs critères ont été adoptés pour l'estimation des mouvements entre blocs. Parmi ces critères, deux sont définis ci-dessous :

- **L'erreur quadratique moyenne (MSE)**

Dans ce cas, le bloc le plus ressemblant est celui pour lequel le MSE est minimale entre le bloc dans la trame courante et le bloc à l'intérieure de la fenêtre de recherche  $W$  dans la trame de référence. Le MSE entre blocs pour les déplacements  $d_x$  et  $d_y$  est définie par :

$$MSE(d_x, d_y) = \frac{1}{M_1 N_1} \sum_{(m,n) \in W} (b(m, n, k) - b(m - d_y, n - d_x, k - 1))^2$$

Où  $M_1 \times N_1$  est La taille du bloc. L'estimation du vecteur de mouvement correspond donc à un vecteur pour lequel le MSE est minimale :

$$\begin{bmatrix} \hat{d}_x \\ \hat{d}_y \end{bmatrix} = \underset{(d_x, d_y)}{\operatorname{arg\,min}} MSE(d_x, d_y)$$

- **L'erreur moyenne absolue (MAD, Mean absolute difference)**

Le nombre d'opérations arithmétique employées dans le critère ci-dessus peut être considérablement réduit si on utilise l'erreur moyenne absolue (MAD). Le MAD entre le bloc courant et celui de référence est définie par :

$$MAD(d_x, d_y) = \frac{1}{M_1 N_1} \sum_{(m,n) \in W} |b(m, n, k) - b(m - d_y, n - d_x, k - 1)|$$

L'estimation du vecteur de mouvement est alors :

$$\begin{bmatrix} \hat{d}_x \\ \hat{d}_y \end{bmatrix} = \underset{(d_x, d_y)}{\operatorname{arg\,min}} MAD(d_x, d_y)$$

Le critère MAD est le plus populaire et adéquat pour l'implémentation VLSI et il est recommandé pour le standard MPEG. Si on ignore le constant de normalisation  $1/M_1 N_1$ , le critère de *Block matching* est appelé *la somme de différence absolue (SAD)*. Ainsi, il est défini par :

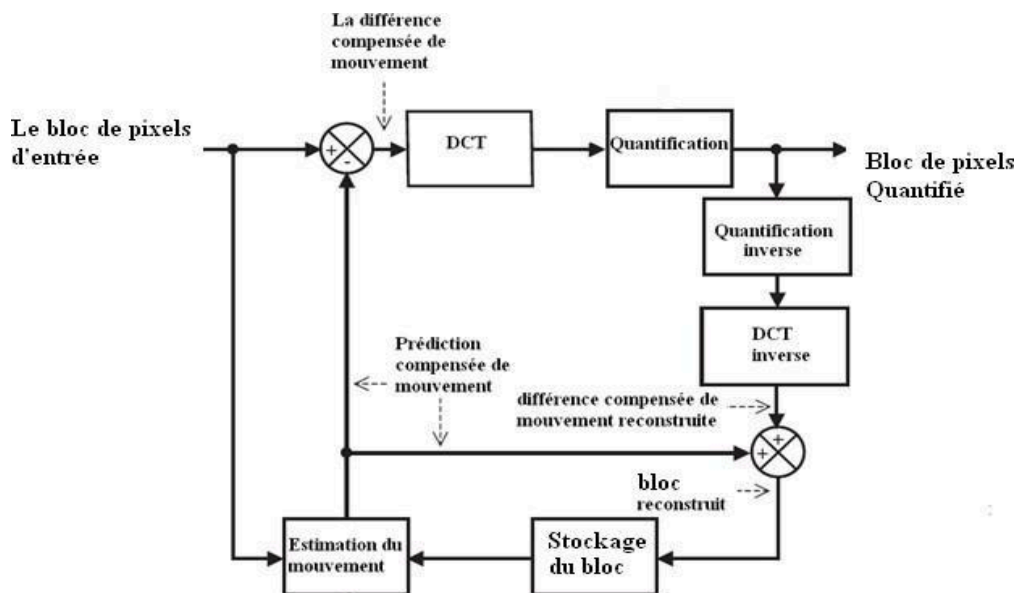
$$SAD(d_x, d_y) = \sum_{(m,n) \in W} |b(m, n, k) - b(m - d_y, n - d_x, k - 1)|$$

### II.3 Compression Vidéo

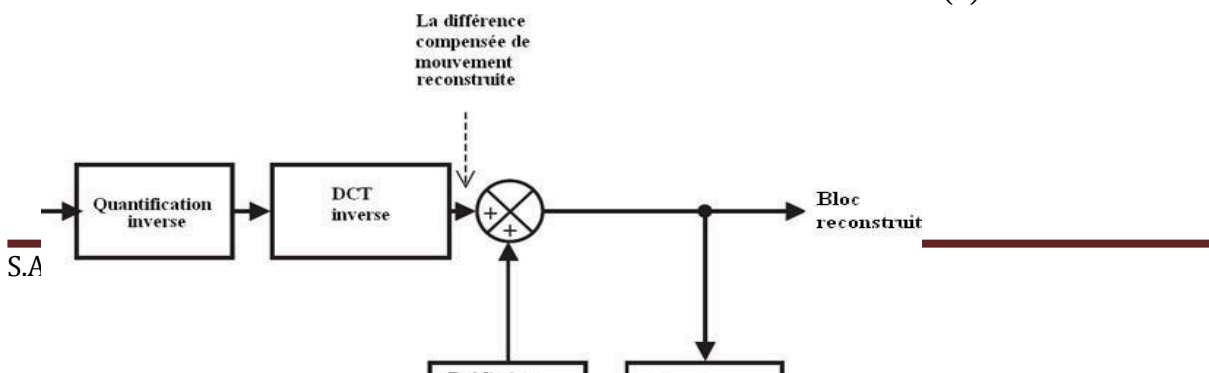
Après avoir étudié la méthode de prédiction avec compensation de mouvement, nous allons décrire maintenant une technique de compression d'image vidéo employée dans la norme MPEG. La figure suivante illustre les principales étapes de cette technique, à savoir :

L'image courante est divisée en blocs rectangulaires. S'il y a des mouvements entre les blocs de l'image courante et ceux de l'image de référence, le vecteur résultant est employé par le prédicteur pour compenser le mouvement et trouver le bloc de prédiction.

La différence entre ce bloc est le bloc actuel produit le bloc différentiel qui est transformé par la DCT (transformée en cosinus discrète) puis quantifier. Les coefficients DCT quantifiés de l'erreur de prédiction compensée de mouvement sont inversement quantifiés et transformés pour reproduire les pixels d'origine. Pour éviter la propagation d'erreurs, le codeur opère dans une boucle fermée (prédiction en boucle fermé). Au décodeur, les blocs sont inversement quantifiés et transformés puis ajoutés au bloc de pixels compensée de mouvement pour produire le bloc reconstruit.



(a)



(b)

*Fig.4. Diagramme en bloc du codage vidéo avec compensation du mouvement  
(a)Codeur, (b) décodeur.*

### III. MANIPULATIONS :

#### **A. Redondance temporelle**

La redondance temporelle identifie les zones redondantes entre les images et ne stocke que les différences. Les zones redondantes sont des reprises des images précédentes. La trame (ou séquence d'image) différentielle aussi appelée résiduelle peut contenir beaucoup moins de détails que la séquence d'image réelle elle-même. En raison de cette réduction de détails dans les trames différentielles, la compression est réalisée. Cette compression est fréquemment appelée *inter-trame*.

#### **Travail demandé :**

1. Extraire le contenu « TP2 MPEG.zip » dans le répertoire de travail.
2. Visualisez sous Matlab les deux séquences d'images 10 et 11.
3. Déterminer l'image différentielle.
4. Visualiser l'histogramme de la trame 11 et celui de la trame différentielle. Comparez ces deux histogrammes ? Conclure.
5. Visualisez les 16 premières séquences d'images. Tracer la courbe de l'évolution de la corrélation temporelle dans cette séquence (utilisez la fonction corr2).

#### **B. Estimation de mouvement**

Pour réaliser une compression beaucoup plus efficace, il est avantageux d'évaluer les mouvements relatifs d'objets entre les trames successives, compenser ces mouvements, puis déterminer la trame différentielle. Ce type de prédiction est connu par prédiction compensée de mouvement.

#### **Travail demandé :**

1. Estimez les vecteurs de mouvements et calculez la trame prédictive compensée de mouvement correspondante entre les deux trames consécutives 10 et 11. (Trame référence et trame courante).
2. Utilisez macro blocs de taille 8\*8 et une fenêtre de taille 16\*16.
3. Visualisez la trame 11 reconstruite.
4. Modifiez les paramètres : taille du macro bloc 16\*16 et de la fenêtre à 32\*32. Quel est leurs effets sur la qualité de prédiction et le coût de codage des vecteurs de mouvements.

### C. Compression Vidéo

Dans cette section, nous allons coder les trames d'une séquence vidéo puis évaluer les performances du codage avec compensation mouvement.

#### Travail demandé :

1. Pour la séquence de trames 10 à 19. Estimez le mouvement de chaque bloc 8×8 à l'intérieure d'une fenêtre de recherche de 16×16 pixels en employant le critère du *block matching* SAD. Choisissez un pas de quantification égal 16.
2. Visualiser les trames originales et celles reconstruites.
3. Tracer la courbe d'évolution du PSNR avec le numéro de la trame (avec compensation de mouvement et sans compensation de mouvement).
4. Conclure.