

Partie 2 - Compression et décompression MPEG

MPEG est une norme de codage d'objets audiovisuels spécifiée par le *Moving Picture Experts Group*. Le codage MPEG prend en compte la corrélation temporelle entre images. Afin d'effectuer la décorrélation temporelle, une différence entre images est effectuée. Cependant, il ne s'agit pas de différence entre deux images successives directement car les objets en mouvement ou le déplacement de la caméra vont entraîner une différence trop importante. C'est pourquoi une estimation de mouvement entre les deux images est effectuée, suivie d'une compensation de ce mouvement pour obtenir une différence d'images la plus faible possible.

Exercice 6 - Estimation et compensation du mouvement

Dans MPEG, l'estimation de mouvement est faite suivant une approche par patches. L'image courante I_C est découpée en macro-blocs de taille 16×16 pixels. Pour chacun de ces macro-blocs, on recherche le macro-bloc le plus proche au sens de l'erreur quadratique moyenne dans l'image de référence I_R . En définissant \vec{v}_C qui contient les coordonnées du premier pixel du macro-bloc dans I_C et \vec{v}_R celles pour le macro-bloc dans I_R , on a le déplacement entre I_C et I_R qui vaut $\vec{v} = \vec{v}_R - \vec{v}_C$. La recherche du macro-bloc optimal répond donc à :

$$\vec{v}^* = \arg \min_{\vec{v}} \left\| \sum_{i,j=1}^{16} I_R \left(\vec{v} + \vec{v}_C + \begin{pmatrix} i \\ j \end{pmatrix} \right) - I_C \left(\vec{v}_C + \begin{pmatrix} i \\ j \end{pmatrix} \right) \right\|^2 \quad (8)$$

Afin de trouver le macro-bloc correspondant dans I_R , on utilise le *Cross Search Algorithm*. Cet algorithme permet de rechercher le macro-bloc optimal suivant un parcours qui diminue l'erreur entre les deux macro-blocs. En se plaçant à la position de départ \vec{v}_C dans I_R , on calcule l'erreur centrale EQM_C entre les deux macro-blocs. Ensuite on va déplacer le macro-bloc de recherche d'un pixel autour de la position \vec{v}_C pour calculer 4 nouvelles erreurs EQM_H , EQM_G , EQM_D , EQM_B (haut, gauche, droite, bas). Si une des erreurs est plus faible que l'erreur au centre, on se déplace vers celle-ci, sinon on s'arrête car on a atteint un minimum local.

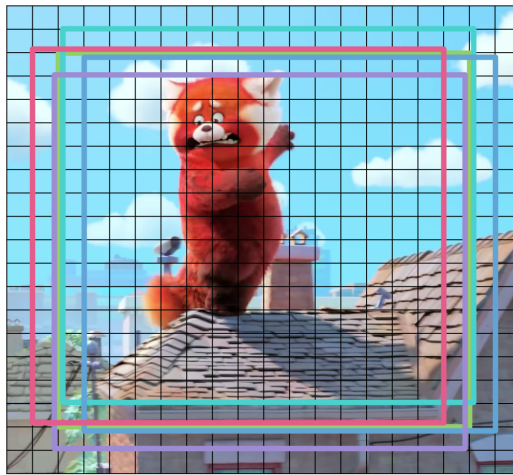


FIGURE 3 – Différents macro-blocs à parcourir lors d'une itération : central (vert), haut (turquoise), gauche (rouge), droite (bleu) et bas (violet).

Complétez la fonction `EstimationMouvement` qui, à partir de l'image courante I_C et l'image de référence I_R calcule la matrice associée aux déplacements relatifs des macro-blocs de I_C . Cette fonction a besoin d'utiliser la sous-fonction `CSAMacroBloc` qui, pour chaque macro-bloc, va parcourir le chemin suivant la diminution de l'erreur quadratique moyenne des différents macro-blocs voisins. Cette sous-fonction a elle-même besoin de la sous-fonction `EQMMacroBlocVoisin` qui calcule l'erreur quadratique moyenne avec les macro-blocs voisins en utilisant la variable `voisin`. **Attention :** Certaines EQM n'ont pas besoin d'être recalculées lorsqu'on se déplace d'un pixel.

Bonus : Utilisez les 8 voisins pour la recherche de minimum au lieu des 4 voisins.

Une fois l'estimation du mouvement effectuée, on souhaite obtenir une prédiction I_P de l'image courante I_C à partir de l'image de référence I_R par compensation du mouvement. Pour ce faire, on remplace chaque macro-bloc de I_C par le macro-bloc de I_R associé au déplacement estimé précédemment. Complétez la fonction [PredictionImage](#) qui crée I_P à partir de I_R et \vec{V} . Ensuite lancez le script [Exercice_6](#) pour tester le bon fonctionnement de l'ensemble.

Exercice 7 - Compression d'une image au format MPEG

Pour la compression MPEG, les images de référence sont compressées au format JPEG pour garder l'information qui sera réutilisée pour les autres images lors de la décompression. Pour l'image courante, ce qui va réellement être codé est la différence (pour décorréler encore et toujours) entre l'image courante I_C et sa prédiction I_P .

Dans l'ordre les étapes de compression dans la fonction [CompressionMPEG](#) sont :

- Calculer les cartes de mouvements \vec{V} des macro-blocs de l'image courante I_C ,
- Calculer la prédiction I_P de l'image courante I_C à partir de l'image de référence I_R et des mouvements \vec{V} ,
- Calculer l'image résiduelle I_{Res} comme la différence entre l'image courante I_C et l'image prédite I_P ,
- Effectuer la compression JPEG I_R^c de l'image de référence I_R (canal [Luminance](#)),
- Effectuer la compression JPEG de l'image résiduelle I_{Res}^c (canal [Residu](#), voir ci-après pour les détails),
- Effectuer le codage entropique des mouvements \vec{V} et ajouter ce poids au poids de l'image résiduelle I_{Res} .

Pour la quantification de l'image résiduelle I_{Res} , la table de quantification $T_{(res)}^q(50)$ est différente. Cette dernière est uniforme pour tous les coefficients (des points en moins pour ceux qui coderont la matrice à la main ...) :

$$T_{(res)}^q(50) = \begin{pmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{pmatrix} \quad (9)$$

Le facteur de qualité F_Q quant à lui reste le même entre l'image de référence et l'image résiduelle. Testez la fonction avec le script [Exercice_7](#).

Exercice 8 - Décompression MPEG et analyse de la qualité de reconstruction

Comme pour JPEG, pour évaluer la qualité de la compression MPEG, il est nécessaire d'effectuer une reconstruction de l'image à partir des coefficients quantifiés. Dans l'ordre les étapes de décompression à compléter dans la fonction [DecompressionMPEG](#) sont :

- Effectuer la décompression JPEG I_R^d de l'image de référence compressée I_R^c (canal [Luminance](#)),
- Calculer la prédiction I_P de l'image courante I_C à partir de l'image de référence I_R^d et des mouvements \vec{V} ,
- Effectuer la décompression JPEG I_{Res}^d de l'image résiduelle I_{Res}^c (canal [Residu](#)),
- Recomposer l'image courante décompressée I_C^d comme la somme entre l'image résiduelle décompressée I_{Res}^d et l'image prédite I_P .

Lancez ensuite le script [Exercice_8](#) qui va effectuer le codage, puis le décodage et afficher le résultat suivant les différents facteurs de qualité F_Q possibles. Diminuez la valeur de la variable [F_Qualite_Max](#) pour que l'exécution s'arrête à un facteur de qualité plus faible et ainsi pouvoir observer les différents artefacts occasionnés par la compression, et la valeur du PSNR associé.