RICM5 TMA

Be - Compression

1.Compression d'images statiques : Etude de la DCT et de ses propriétés.

La compression d'image fixe utilise très souvent la transformée discrète en cosinus (DCT). (en particulier pour Jpeg)

Equation de la DCT :

$$F(u,v) = \alpha(u)\alpha(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j)\cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2j+1)v\pi}{2N}\right]$$ pour i et j : 0..N-1 avec

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{si u = 0} \\ \sqrt{\frac{2}{N}} & \text{ailleurs} \end{cases}$$

où N est la taille du bloc de pixels considéré. Il est possible de faire la DCT sur l'image en entier, mais en pratique, l'image est découpée en blocs égaux de taille N*N et la DCT est appliquée sur chacun de ses blocs.

La DCT inverse est donnée par l'équation suivante :

$$I(i,j) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2j+1)v\pi}{2N}\right]$$

Un premier outil d'évaluation d'une méthode de compression est très souvent le PSNR, qui lui, donne une évaluation complète de la chaîne de codage. En effet, on calcule l'erreur quadratique moyenne entre l'image originale et l'image obtenue après compression.

PSNR : $10\log_{10}\left(255^2 / EQM\right)$ en dB

avec l'erreur quadratique moyenne : $EQM = \frac{1}{N*M}\sum_{ij}\left(I_{ij} - \hat{I}_{ij}\right)^2$ avec I l'image originale et Î l'image compressée puis décompressée, N*M la taille de l'image.

Les fonctions dct2 et idct2 sont fournies dans la librairie dct2.py.

Principe de la compression jpeg

- Réalisez un programme qui calcule la DCT d'une image. Vous pouvez dans un premier temps prendre toute l'image dans un seul bloc. Visualisez l'image DCT, quelle conclusion pouvez-vous faire sur la localisation et la répartition des données? Où se trouve l'essentiel de l'information.

- Réalisez une fonction qui découpe votre image en bloc n*n puis calcule la DCT sur tous les blocs. Entrée : l'image. Sortie : Les blocs n*n.

- Programmez la table de quantification suivante :
  Q(i,j)=1+((1+i+j))*compression,

Pour un bloc n*n la table doit être de taille n*n. On choisira de fixer n=8.
Quantifier ensuite vos blocs avec cette table : Fq(u,v)=np.round(F(u,v)/Q(u,v))

- Compression-decompression.

On ne va pas dans ce TP étudier les méthodes de codage canal aussi on va passer directement à la phase de décompression

Réalisez la dé-quantification (pour ceci on suppose que le décodeur connaît la table de quantification), puis la DCT inverse pour obtenir en sortie l'image reconstruite.

Etudiez les cas n=8, n=16, n=4. D'après vous pourquoi le groupe jpeg a choisi la taille 8 ?

Etude des métriques de qualité

Pour aller plus vite, on ne va plus faire toute la chaine DCT avec la quantification etc…On utilisera dans la suite les fonctions de lecture et d'écriture de fichiers jpeg qui permettent de compresser, décompresser une image puis de récupérer les différentes informations du fichier, dont sa taille (voir le fichier python d'exemple).

- Etudiez l'influence du facteur « qualité » : calculez le PSNR. Quel est pour vous le seuil critique à ne pas dépasser pour la qualité ? Tracez les courbes PSNR=f(qualité) et PSNR=f(taux de compression)

- Etude d'un autre indicateur de qualité pour image avec référence : le SSIM de Zhou Wang & Bovik

  Les fonctions de qualité sont données dans le module measure de skimage et les explications dans l'article en annexe.
  Implémentez cet indicateur de qualité pour différentes compression d'une image. Tracez la courbe SSIM=f(taux de compression), et SSIM=f(qualité). Pour ceci prenez une image BMP puis avec un compresseur quelconque JPEG (dont celui de PIL '*.Image.save()') compressez-là à différents taux. Sur chacune de ces images ainsi obtenues faites le calcul du PSNR et du SSIM. En déduire les courbes.
  Conclusions sur ces deux indicateurs ?

2. La Boucle H261 ou/et MPEG1/2

Sur un couple d'images successives d'une séquence In et In-1, Réalisez un programme qui fera la prédiction de In à partir d'une estimation de mouvement entre In et In-1. La méthode utilisée sera celle du block-matching. Les blocs de taille 16*16 effectueront une recherche exhaustive. Les vecteurs de mouvement seront restreints à des déplacements de [-15;15],

autour du pixel traité, les pas seront entiers. Pour réaliser la méthode de prédiction au niveau du codeur, pour prédire chaque bloc, il il faut d'abord implémenter la méthode d'estimation de mouvement (algorithme dans le poly) puis appliquer la prédiction grâce au mouvement obtenu.

Complétez votre boucle de compression en introduisant la DCT qui devra réduire les redondances spatiales de vos images, réalisez le programme qui simule la boucle de la figure 2, où T est la DCT.

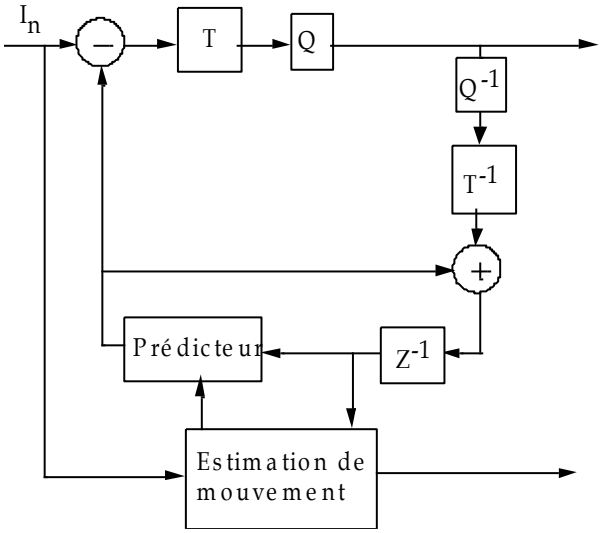

Fig. 1 Boucle de type H261

Mettez en évidence la divergence d'une telle boucle (en utilisant toute la séquence). Pour vous quelle est dans votre cas la fréquence de rafraichissement à choisir ? Illustrez vos conclusions et résultats.