

Travail pratique n° 5

Compression d'images

Instructions

- Les travaux pratiques peuvent être effectués seul ou par équipe de deux *maximum*. La composition de chaque équipe doit être indiquée au professeur au plus tard lors de la séance.
- Le compte rendu doit comporter une réponse concise mais complète à chacune des questions, accompagnée au besoin des courbes, figures et images appropriées.
- Le compte rendu peut prendre deux formes : soit un fichier pdf accompagné des scripts et fonctions Matlab que vous avez développés, soit un document (pdf ou html) généré à l'aide des fonctionnalités de publication de Matlab (menu “File / Publish” de l'éditeur Matlab), accompagné de son fichier source Matlab. Dans tous les cas, l'ensemble des fichiers doit être placé dans une unique archive **zip**.
- Le compte rendu doit être remis au plus tard 7 jours après la séance en utilisant l'outil approprié disponible sur le site web du cours.
- Le travail doit être remis par un seul des membres du groupe. Si tel n'est pas le cas, la version la plus récente du travail remis est prise en compte.

1 Introduction

Ce travail pratique a pour objet le développement et la mise en œuvre d'une technique de codage et décodage d'images inspirée de la norme JPEG, et de la comparer à un codage JPEG standard.

Les fonctions Matlab et les données à utiliser pour effectuer ce travail pratique se trouvent dans l'archive **TP5.zip** disponible sur le site web du cours.

2 Définition de la norme EPEG

L'objet du travail pratique est de d'étudier et de mettre en œuvre les principales étapes du codage et du décodage JPEG, à l'exception du codage / décodage des symboles qui sera remplacé par un codage LZW effectué par le programme **zip**. On définit donc une norme, que nous baptiserons EPEG pour *ELE8812 Photographic Experts Group*, qui a les caractéristiques suivantes :

- codage d’images monochromes de taille maximale $2^{16} \times 2^{16}$ pixels, représentés sur 256 niveaux ;
- compression par transformée en cosinus discrète (TCD) 2D, sur des blocs de taille 8×8 pixels, après décalage de l’image pour l’amener entre -128 et 127 ;
- quantification des coefficients de la TCD de chaque bloc selon la table de normalisation JPEG ;
- possibilité d’appliquer à la table de normalisation un coefficient multiplicatif α compris entre 10^{-1} et 10 , afin de pouvoir faire varier le taux de compression ;
- ordonnancement des coefficients quantifiés de la TCD de chaque bloc selon le schéma « zig-zag » de la norme JPEG ;
- application d’un codage par longueur de plage sur chaque suite de coefficients quantifiés et réordonnés.

Une image complètement codée sera composée des éléments suivants :

- le nombre de M de lignes et le nombre N de colonnes de l’image, codés chacun sur un entier de 16 bits non signé (`uint16`) ;
- le coefficient α , représenté par un entier signé n tel que $\alpha = 10^{n/128}$. n est codé sur un entier signé de 8 bits (`int8`) ;
- Suite des coefficients de la TCD de chaque bloc, codés comme suit :
 - représentation des coefficients par un entier signé compris entre -127 et 127 (`int8`) ;
 - représentation de toute plage de deux coefficients nuls ou plus par le code -128 suivi d’un entier signé (`int8`) ayant pour valeur le nombre de valeurs nulles consécutives ;
 - code « fin de bloc » constitué par deux valeurs -128 consécutives (`int8`) placées immédiatement à la suite du dernier coefficient non nul du bloc ;
 - balayage des blocs *colonne par colonne*.

3 Travail demandé

3.1 Codage et décodage EPEG (12 points)

On demande d’écrire deux fonctions Matlab ayant les fonctionnalités suivantes :

Codage

```
function imc = EPEGc(im, alpha)
%
% Fonction permettant le codage "EPEG" d'une image à 256 niveaux de gris
%
% Paramètres d'entrée
%   im : image à coder, de type uint8
%   alpha : facteur multiplicatif de la table de normalisation JPEG,
%   compris entre 10^(-1) et 10
%
% Paramètre de sortie
%   imc : image codée. Structure comprenant les champs suivants
%       M : nombre de lignes de l'image. uint16
%       N : nombre de colonnes de l'image. uint16
```

TABLE 1 – Fonctions Matlab utiles pour résoudre la question 3.1

Type de traitement	Fonctions
Traitement d'images bloc par bloc	blkproc ou blockproc
Transformée en cosinus 2D directe et inverse	dct2, idct2

```
% Facteur : entier compris entre -128 et 127 codant alpha selon une
% échelle logarithmique. int8
% Code : suite des codes. Tableau int8 1D de taille variable
%
```

Décodage

```
function imd = EPEGd(imc)
%
% Fonction permettant le décodage "JPEG" d'une image à 256 niveaux de gris
%
% Paramètre d'entrée
% imc : image à décoder. Structure comprenant les champs suivants
% M : nombre de lignes de l'image. uint16
% N : nombre de colonnes de l'image. uint16
% Facteur : entier compris entre -128 et 127 codant alpha selon une
% échelle logarithmique. int8
% Code : suite des codes. Tableau int8 1D de taille variable
%
% Paramètre de sortie
% imd : image décodée. Tableau uint8 de taille (M, N)
%
```

Pour cela, il est recommandé d'utiliser les fonctions données au tableau 1. Par ailleurs, il est demandé d'étendre au besoin l'image par des zéros afin que ses dimensions soient multiples de 8. La partie étendue devra être supprimée au décodage. Enfin, l'archive TP5.zip fournit les fonctions zigzag.m et dezigzag.m qui permettent de convertir un bloc de taille 8×8 en un vecteur colonne de taille 64 selon le schéma de la norme JPEG et réciproquement, ainsi que le fichier Matlab EPEG_Q.mat qui contient la matrice de quantification de coefficients de TCD à utiliser. La fonction compjpg, qui compresse une image selon la norme JPEG, fait également partie de l'archive TP5.zip.

Une liste de fonctions Matlab utiles pour résoudre cette question est donnée au tableau 1.

3.2 Test du codeur - décodeur (8 points)

Le codeur / décodeur sera testé à l'aide des images cameraman.tif et Paolina.gif incluses dans l'archive TP5.zip. En faisant varier le paramètre α , on tracera l'erreur quadratique entre image codée et l'image originale en fonction de la taille de l'image codée. Celle-ci sera évaluée de deux manières (4 points) :

1. par le nombre d'octets de la variable imc retournée par la fonction EPEGc ;

2. par le nombre d'octets de cette même variable `imc`, stockée dans un fichier Matlab compressé par `zip`.

Ces courbes seront comparées à celles obtenues pour un codage JPEG (utilisez la fonction `compjpg` fournie dans l'archive `TP5.zip` pour effectuer ce codage) (2 points). Vous observerez également l'effet de la compression par zip sur fichiers images originaux et JPEG (2 points).

Remarque Les fichiers Matlab `.mat` peuvent être stockés sous forme compressée ou non. Pour évaluer correctement l'effet des différentes étapes du codage, il faut choisir un format de fichiers `.mat non compressé`. Ce choix peut être fait dans le menu de configuration de Matlab, à la section « General / Mat-Files », ou en utilisant l'option `-v6` de l'instruction `save`.