

ESIR, 2^e année

TP 2 - La transformée en cosinus discrète 2D

O. Le Meur

Mots clés : OpenCV, DCT, erreur quadratique moyenne, PSNR, entropie.

1 Introduction

1.1 Objectifs du TP

Ce TP a plusieurs objectifs :

- prise en main de la transformée en cosinus discrète 2D sur des images naturelles ;
- implémentation d'une transformée par bloc 8×8 .

1.2 Notation

Le tp sera noté en fonction des critères suivants :

- La qualité du compte-rendu au format pdf (**le nom du pdf sera composé des noms des auteurs**). Un bon compte-rendu comporte une introduction, une conclusion (voire une discussion) et les réponses aux questions. Un tableau de valeur sans commentaire est inutile, et pas noté. La capacité à critiquer et à analyser les résultats est fondamentale. Faites preuve de sens critique (à bon escient) ;
- La qualité de l'écriture des programmes et la facilité à les lire-comprendre ;
- Le compte rendu doit être rendu à la date indiquée.

1.3 Matériel nécessaire

Le matériel nécessaire est le suivant :

- les fonctions que vous avez développées lors du premier TP ;
- quelques images de test ont été ajoutées dans le répertoire image.

1.4 Une vue générale

Votre travail consiste à étudier les performances de la DCT en terme de compression d'images.

- appliquer la DCT sur une image ou par bloc ;
- étudier la répartition et l'influence des coefficients DCT sur la qualité globale ;
- calculer l'entropie des images avant et après compression ;
- calculer le PSNR de l'image après DCT inverse (avec et sans annulation de coefficients) ;
- visualiser les histogrammes des coefficients DCT.

2 Etude de la DCT appliquée sur une image

2.1 DCT et DCT inverse

Question 2.1. Mettre en place un programme qui ouvre une image et la convertit en YCrCb. Appliquer la DCT sur chacune des composantes.

Nous vous invitons à transformer votre variable de type Mat composée de 3 canaux en un vector < Mat > en utilisant la commande split(...). Appliquer la fonction dct() sur chaque composante.

Question 2.2. Appliquer la DCT inverse et vérifier sa réversibilité en calculant le PSNR entre l'image source et l'image après transformation.

Il sera suffisant de calculer le PSNR sur la composante Y.

2.2 Répartition des coefficients DCT

Question 2.3. Visualiser les coefficients DCT pour les trois canaux et commenter la répartition. Vous pourrez prendre des images bien particulières pour étayer votre argumentation.

Appliquer une compression logarithmique afin de visualiser les coefficients (attention, les coefficients sont signés). Par exemple, si la variable dctCoeff contient les coefficients DCT pour les trois composantes, on pourra appliquer la formule suivante :

$$dctCoeff[k].at < float > (i,j) = \log(1 + \text{fabs}(dctCoeff[k].at < float > (i,j))) / \log(1 + \text{maxValLoc}) * 255; \quad (1)$$

Il sera alors possible d'afficher dctCoeff[k] avec la fonction imshow(). A noter qu'on peut appliquer une color map, applyColorMap(in,out, COLORMAP_JET); attention in et out sont au format CV_8UC1. Un exemple est donné à la figure 1.

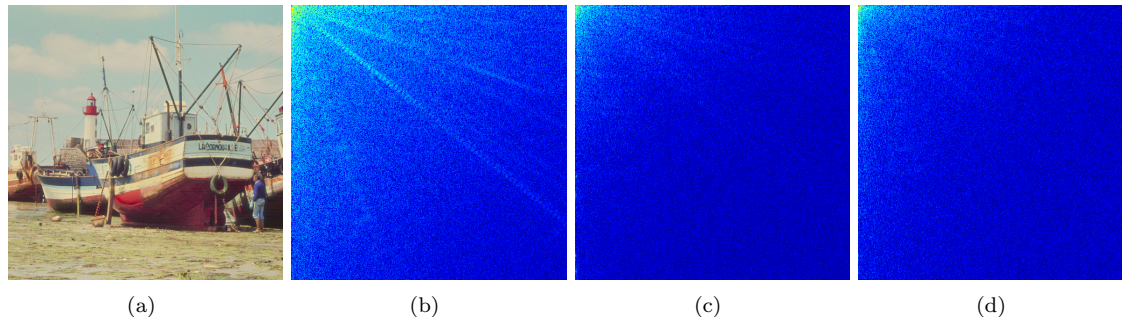


FIGURE 1 – Répartition des coefficients DCT pour l'image *boats* (a) : (b) composante Y, (c) composante Cr, (d) composante Cb. (colormap=COLORMAP_JET)

Question 2.4. Visualiser l'histogramme des coefficients DCT et calculer son entropie. Vous comparerez cette entropie à l'entropie de l'image source. Vous commenterez l'allure des histogrammes (image originale et image DCT) et les valeurs d'entropie.

2.3 Influence des coefficients DCT sur la qualité de l'image

Question 2.5. Afin de mettre en évidence l'influence des coefficients DCT, on souhaite annuler un certain nombre de valeurs. Des exemples de masques binaires sont donnés à la figure 2 : on conserve la partie blanche et on annule la partie noire.

- Calculer le PSNR et la carte de distortion pour différents masques (on travaillera sur la composante Y).
- Que se passe-t-il si on annule des coefficients de très basses fréquences, et notamment le coefficient de fréquence nulle (illustrer votre réponse, montrer l'image reconstruite et donner son PSNR).

OpenCV permet d'agir très facilement sur une zone de l'image. Par exemple, $\text{dctDCT}[k](\text{Rect}(x, y, \text{myWidth}, \text{myHeight})) = 0$; annule les valeurs de la zone rectangulaire définie.

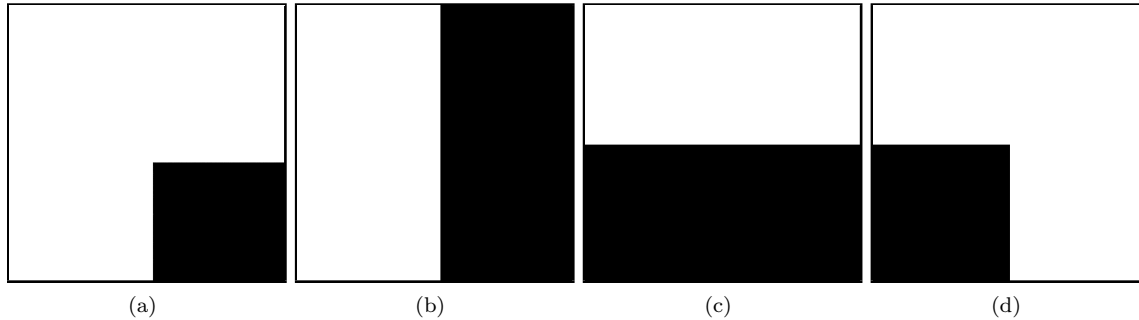


FIGURE 2 – Exemples de masque à utiliser

Question 2.6. Sommes-nous dans un contexte de transformation globale ? Si oui, pourquoi ? Illustrer vos arguments.

3 Etude de la DCT appliquée par bloc 8×8

Dans cette partie, nous nous intéressons à réaliser la transformation DCT sur des blocs de taille 8×8 (on pourra utiliser l'image *boat*).

Question 3.7. Mettre en place la transformation DCT et DCT inverse des blocs 8×8 . Vérifier la réversibilité.

On pourra utiliser la méthode suivante pour extraire un bloc de l'image `Mat block = img(Rect(j,i,8,8))`;

Question 3.8. On se propose d'appliquer sur les coefficients transformés $T(u, v)$ les matrices de la figure 3 :

- Pour chaque bloc, appliquer la matrice de la figure 3 (a). On conserve les coefficients marqués par la valeur 1. Appliquer la DCT inverse et mesurer le PSNR ;
- Pour chaque bloc, appliquer la matrice de la figure 3 (b) : $\hat{T}(u, v) = \lfloor \frac{T(u, v)}{Z(u, v)} \rfloor$, $\lfloor * \rfloor$ représente un arrondi au plus proche voisin. Appliquer la DCT inverse (attention, il faut multiplier les coefficients $\hat{T}(u, v)$ par $Z(u, v)$ avant d'effectuer la transformée) et mesurer le PSNR.

<pre> 1 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 </pre>	<pre> JPEG Threshold Matrix Z(u,v) 16 11 10 16 24 40 51 61 12 12 14 19 26 58 60 55 14 13 16 24 40 57 69 56 14 17 22 29 51 87 80 62 18 22 37 56 68 109 103 77 24 35 55 64 81 104 113 92 49 64 78 87 103 121 120 101 72 92 95 98 112 100 103 99 </pre>
(a)	(b)

FIGURE 3 – Exemples de masque à utiliser : (a) masque binaire simple ; (b) matrice de pondération de la norme JPEG.