Rapport de TP - SY26

TP04 - La compression JPEG

Rémi Burtin

Cyril Fougeray

28 mai 2014

Université de Technologie de Compiègne

1 Introduction

Le but de ce TP est de mettre en œuvre certaines étapes de l'algorithme de compression JPEG.

2 Transformée en cosinus discrète (DCT)

2.1 Mise en œuvre de la fonction MyDCT

Cette première fonction nous permet de calculer la transformée en cosinus discrète d'un bloc de taille 8x8.

Pour cela, nous utilisons les formules suivantes :

$D = X_m B Y_m^T$

avec $X_m(u,x) = \frac{1}{2}C(u)cos(\frac{(2x+1)\pi u}{16})$ et $Y_m(v,y) = \frac{1}{2}C(v)cos(\frac{(2x+1)\pi v}{16})$

en prenant $C(0) = \frac{1}{\sqrt{2}}$ et $C(k) = 1$ pour $k \in [1..7]$.

Les coefficient u et v varie de 0 à 7, donc les matrices X_m et Y_m ont 8 lignes. Par ailleurs, la matrice B avec laquelle nous travaillons est une matrice 8x8. Ainsi, les coefficients x et y sont pris entre 1 et 8. On obtient donc $X_m = Y_m$.

Nous calculons les matrices X_m et Y_m en remplaçant les 4 variables (u, v, x, y) dans les formules, nous multiplions les matrices X_m, B et Y_m et obtenons ainsi la matrice D, la transformée en cosinus discrète de B.

Le code de la fonction est en annexe A.1.

2.1.1 Résultats

Après avoir exécuté la fonction, nous nous rendons compte du résultat :

```
>> Bref=[ 139 144 149 153 155 155 155 155;
144 151 153 156 159 156 156 156;
150 155 160 163 158 156 156 156;
159 161 162 160 160 159 159 159;
159 160 161 162 162 155 155 155;
161 161 161 161 160 157 157 157;
162 162 161 163 162 157 157 157;
162 162 161 161 163 158 158 158];
```

```
>> BrefDCT = MyDCT(Bref)

BrefDCT =

   1.0e+03 *

 1.2596   -0.0010   -0.0121   -0.0052    0.0021   -0.0017   -0.0027    0.0013
-0.0226   -0.0175   -0.0062   -0.0032   -0.0029   -0.0001    0.0004   -0.0012
-0.0109   -0.0093   -0.0016    0.0015    0.0002   -0.0009   -0.0006   -0.0001
-0.0071   -0.0019    0.0002    0.0015    0.0009   -0.0001   -0.0000    0.0003
-0.0006   -0.0008    0.0015    0.0016   -0.0001   -0.0007    0.0006    0.0013
 0.0018   -0.0002    0.0016   -0.0003   -0.0008    0.0015    0.0010   -0.0010
-0.0013   -0.0004   -0.0003   -0.0015   -0.0005    0.0017    0.0011   -0.0008
-0.0026    0.0016   -0.0038   -0.0018    0.0019    0.0012   -0.0006   -0.0004
```

La matrice $BrefDCT$ correspond au résultat souhaité.

3 Quantification

Cette partie nous permet de calculer une matrice de quantification, qui nous permettra ensuite de quantifier le bloc après la DCT. Le bloc quantifié s'obtient en divisant chacun des coefficients après la transformée en cosinus discrète par le facteur de quantification situé à la même position dans la matrice de quantification QM :

$$Dq(i,j) = \left\lfloor \frac{D(i,j)}{QM(i,j)} + 0.5 \right\rfloor$$

L'oeil humain étant sensible aux fréquences basses (situées dans le coin supérieur gauche de la matrice après DCT), nous devons diviser ces coefficients par un facteur faible. Voici la matrice utilisée pour un facteur $Quality = 50$:

$$\text{QM} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} * \text{Fq}$$

Le facteur *Quality* fourni à la fonction en paramètre d'entrée est compris entre 1 (très mauvais) et 100 (très bien). Si $Quality = 50$ alors $Fq = 1$. Afin de trouver la fonction $f()$ prenant comme paramètre *Quality*, nous avons recherché du côté des standards JPEG qui nous dit :

$$Fq(Quality) = \begin{cases} (100 - Quality)/50 & Quality \geq 50 \\ 50/Quality & \text{sinon} \end{cases}$$

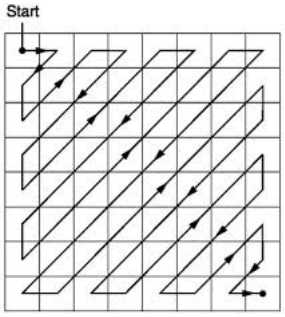
Nous obtenons ainsi Fq. La matrice de quantification est alors modifiée en conséquence. Il faut faire attention à ce que les valeurs de la matrice soit compris entre 1 et 255. Si certaines valeurs sont plus petites (*resp.* plus grandes) nous les fixons à 1 (*resp.* à 255). Pour cela, nous utilisons la fonction *find*, cf Annexe A.2.

Nous avons maintenant la matrice de quantification, nous pouvons alors calculer Dq, le bloc quantifié après la DCT :

```
Dq = round(BrefDCT./QM)
```

4 Parcours en zigzag

Afin de pouvoir compresser efficacement l'information du bloc Dq on se propose de transformer la matrice 88 en un vecteur de longueur 64 en parcourant le bloc dans un ordre qui favorise la concentration des coefficients nuls à la fin du vecteur.



Pour cela, nous avons à disposition un vecteur contenant les indexes des positions successives du parcours en zigzag dans le bloc 8x8. Matlab nous permet facilement de récupérer le vecteur à partir ce des indexes :

```
%% Le parcours en ZigZag (Indexes)
zig=[1 9 2 3 10 17 25 18 ...
     11 4 5 12 19 26 33 41 ...
     34 27 20 13 6 7 14 21 ...
```

```
     28 35 42 49 57 50 43 36 ...
     29 22 15 8 16 23 40 37 ...
     44 51 58 59 52 45 38 41 ...
     24 32 38 46 53 60 61 54 ...
     47 40 48 55 62 63 56 64];

Vzig=Dq(zig);
```

5 Codage

6 Codage/décodage d'une image

7 Conclusion

A Codes source MATLAB

A.1 Transformée en cosinus discrète d'un bloc 8x8

```
function D = MyDCT(B)
X = zeros(8,8);
Y = zeros(8,8);
C = [1/sqrt(2) 1 1 1 1 1 1 1];

for i=0:7,
    for j=0:7,
        X(i+1,j+1) = C(i+1)/2*cos(((2*(j)+1)*pi*(i))/16);
    end;
end;

Y = X;
D = X * B * Y';

end
```

A.2 Quantification

```
function QM = QuantM(Quality)
    %Cf standards JPEG
    if Quality >= 50
        Fq = (100 - Quality)/50;
    else
        Fq = 50/Quality;
    end;

    QM = round([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] * Fq);

      %On borne les valeurs a 255
      index = find(QM > 255);
      QM(index) = 255;

      index = find(QM == 0);
      QM(index) = 1;
end
```