

# TERI : Traitement et Reconnaissance d'images

## Cours Master 2 IAD

Isabelle Bloch – ENST / TSI

Florence Tupin – ENST / TSI

Antoine Manzanera – ENSTA / UEI



# Codage d'image :

## Exemple de la compression *Jpeg*

- Représentation fréquentielle et DCT
- Quantification des coefficients DCT
- Parcours des blocs DCT et codage par plage
- Codage entropique et algorithme de Huffman

La norme de compression Jpeg utilise une représentation des images sous forme de *transformée en cosinus discret bidimensionnel* (DCT2d) :

**Transformée directe :**

$$C(u, v) = \frac{2}{n} \left( \frac{1}{\sqrt{2}} \right)^{\delta_{(0,0)}(u,v)} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} I(x, y) \cdot \cos \pi \frac{(2x+1)u}{n} \cdot \sin \pi \frac{(2y+1)v}{n}$$

**Transformée inverse :**

$$I(x, y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} \left( \frac{1}{\sqrt{2}} \right)^{\delta_{(0,0)}(u,v)} C(u, v) \cdot \cos \pi \frac{(2x+1)u}{n} \cdot \sin \pi \frac{(2y+1)v}{n}$$

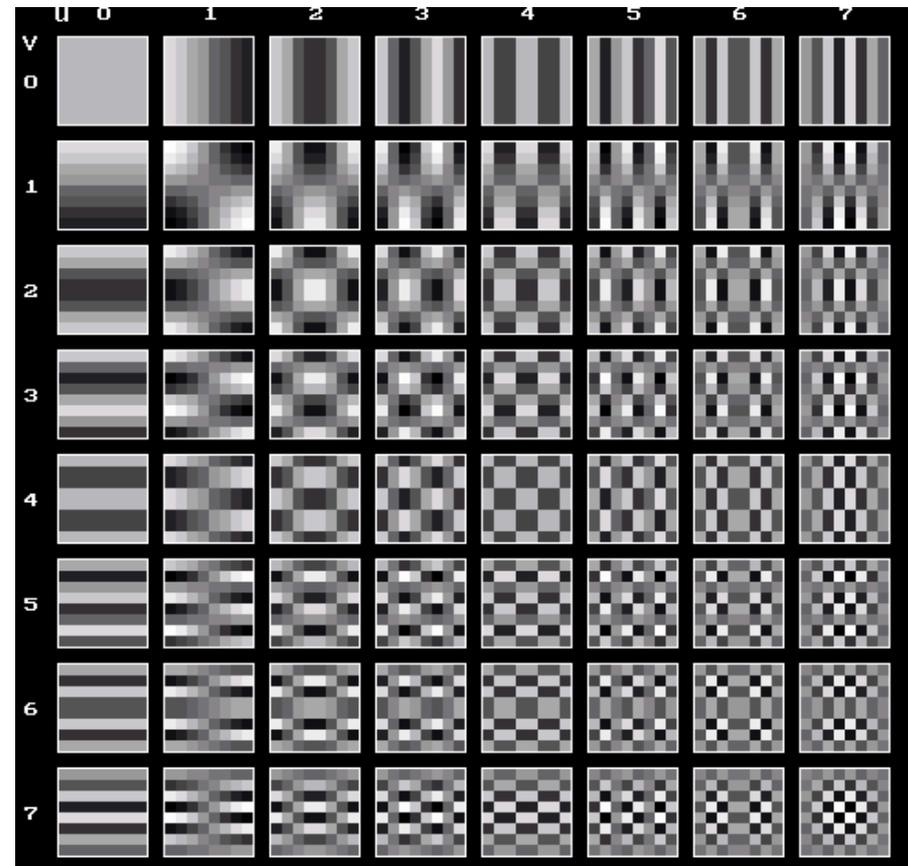
Dans cette représentation, un bloc (image) de taille  $n \times n$  est représenté par ses composantes dans une base de  $n \times n$  fonctions élémentaires  $F_{u,v}(x, y)$  :

$$I(x, y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} C(u, v) \cdot F_{u,v}(x, y)$$

avec :  $F_{u,v}(x, y) = \left( \frac{1}{\sqrt{2}} \right)^{\delta_{(0,0)}(u,v)} \cos \pi \frac{(2x+1)u}{n} \cdot \sin \pi \frac{(2y+1)v}{n}$

L'avantage de cette base, est que contrairement à la base canonique, certaines composantes peuvent être considérées comme moins importantes. C'est cette propriété qu'exploite la compression avec perte.

Ci contre, la base DCT2d pour des blocs de taille  $8 \times 8$  :



Les 64 fonctions élémentaires  $F_{u,v}$  ( $n = 8$ )

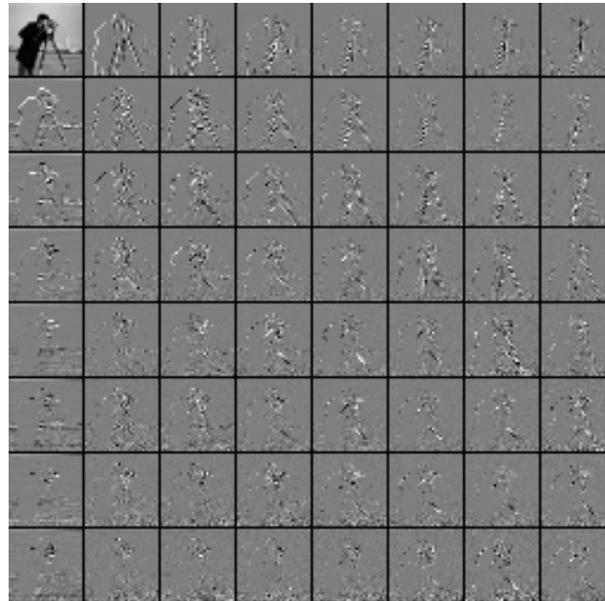
Sur de grandes images, cette transformation est appliquée à *tous les blocs carrés d'un découpage régulier*, voir exemple ci-contre ( $n = 8$ ) :

Le mode de représentation (1) de la DCT2d préserve la localisation des blocs de l'image originale : il illustre bien *l'irrégularité spatiale de la répartition de l'information*.

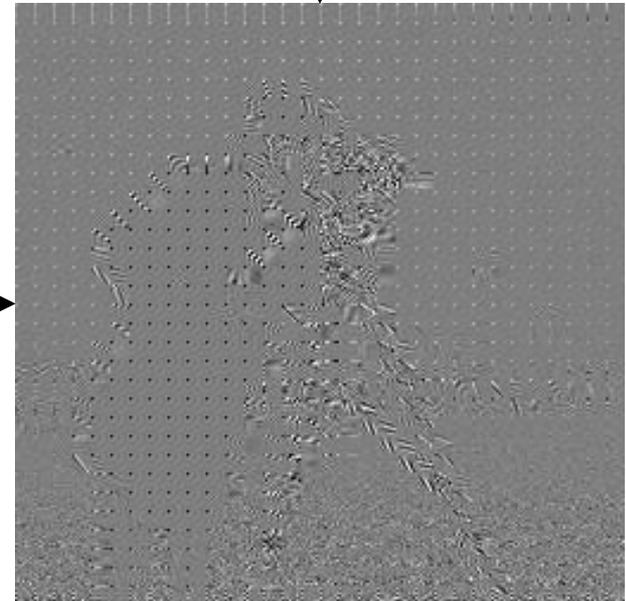
Le mode de représentation (2) de la DCT2d regroupe les pixels de tous les blocs par coefficient. Il permet de visualiser le *découpage en bande de fréquence*, et d'apprécier la différence de leur *pertinence* d'un point de vue *psychovisuel*.



La compression Jpeg exploite cette différence en *quantifiant plus fortement*, voire en supprimant les *fréquences les plus élevées* de la DCT2d



(2)



(1)

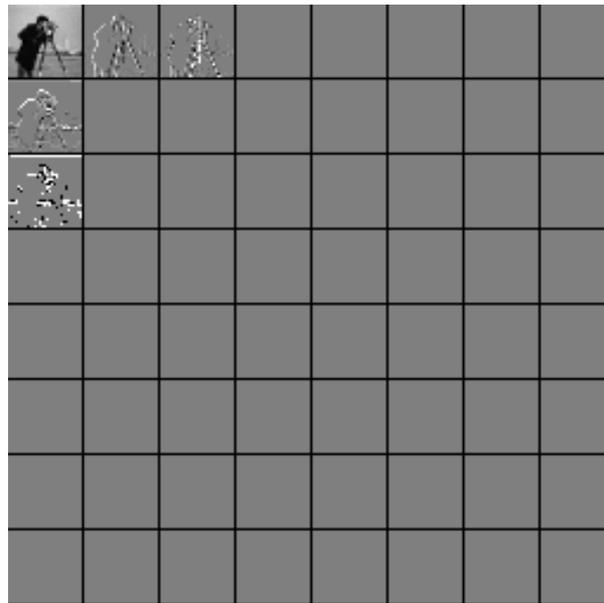
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 0,25  
Size : 16 Ko  
Mean Square Error : 378,7

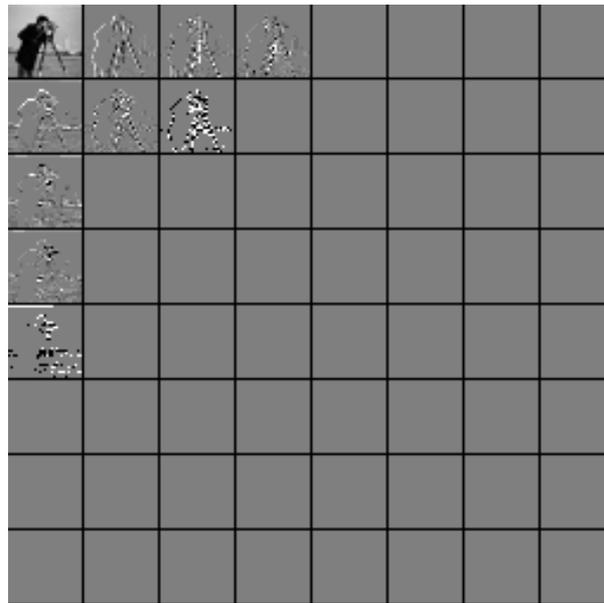
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 0,5  
Size : 32 Ko  
Mean Square Error : 239,5

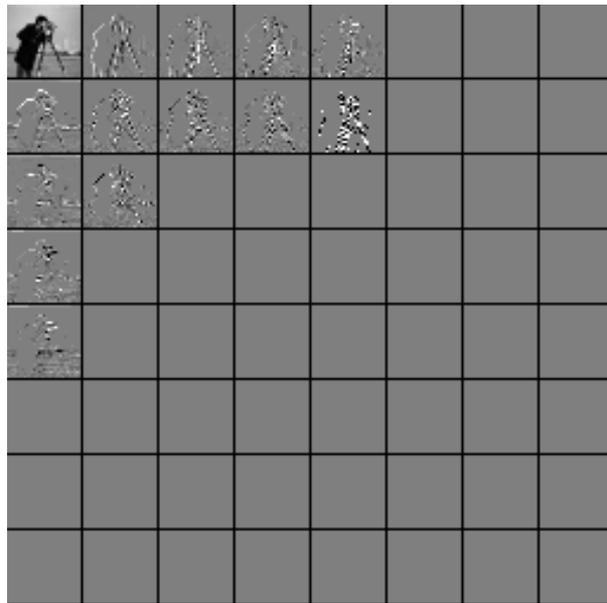
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 0,75  
Size : 48 Ko  
Mean Square Error : 171,8

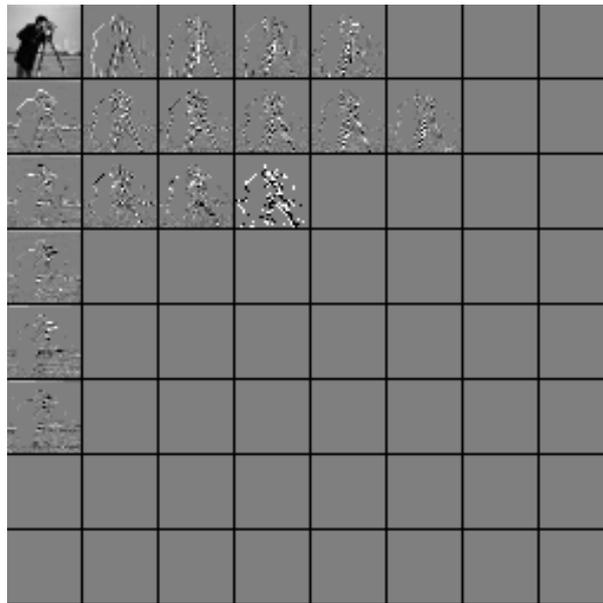
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 1  
Size : 64 Ko  
Mean Square Error : 130,2

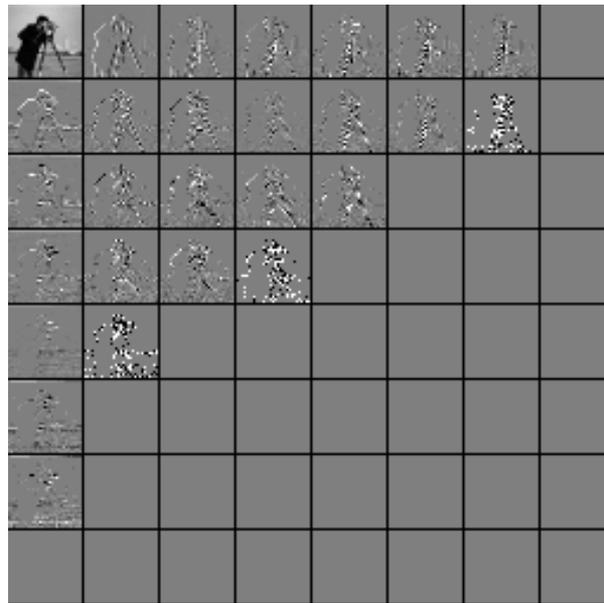
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 1,5  
Size : 96 Ko  
Mean Square Error : 79,8

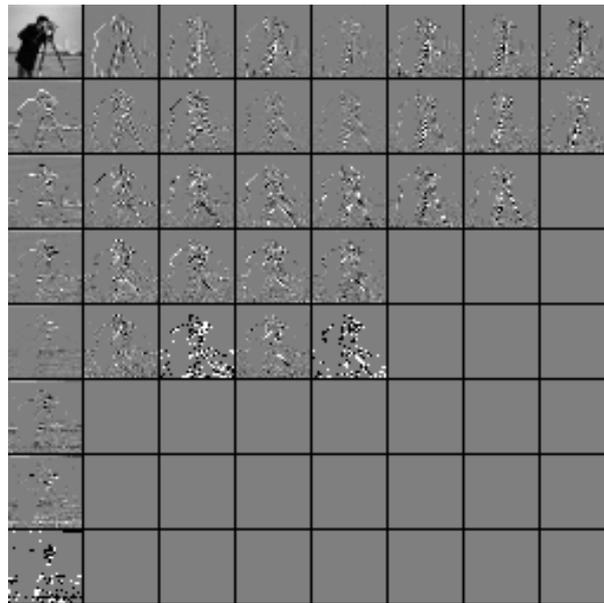
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 2  
Size : 128 Ko  
Mean Square Error : 50,0

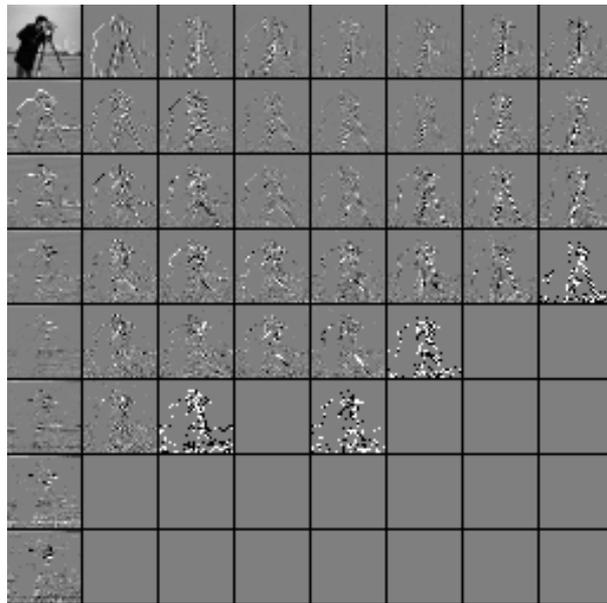
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 2,5  
Size : 160 Ko  
Mean Square Error : 31,6

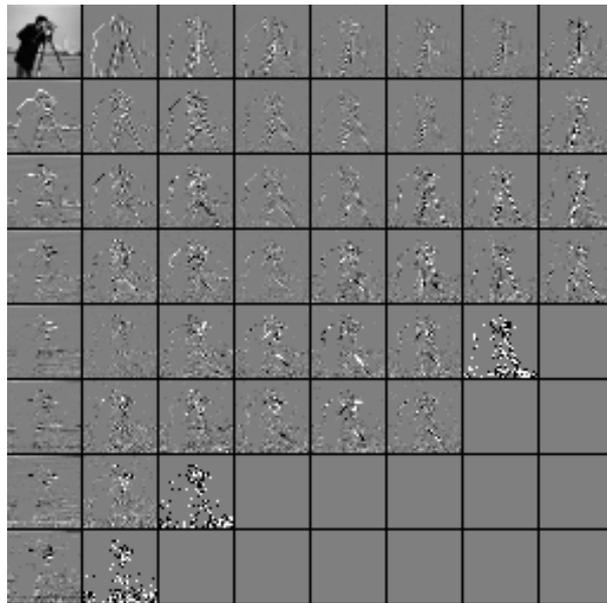
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 3  
Size : 192 Ko  
Mean Square Error : 19,9

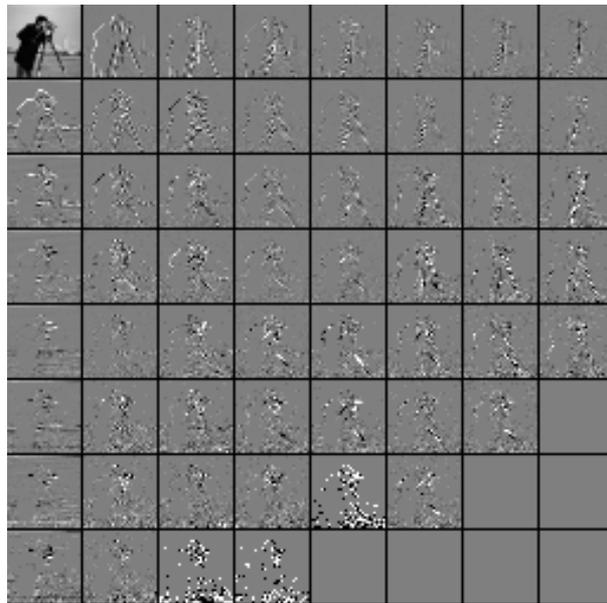
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*



256x256 pixels codés  
sur 8 bits.

Bitrate : 3,5  
Size : 224 Ko  
Mean Square Error : 12,4

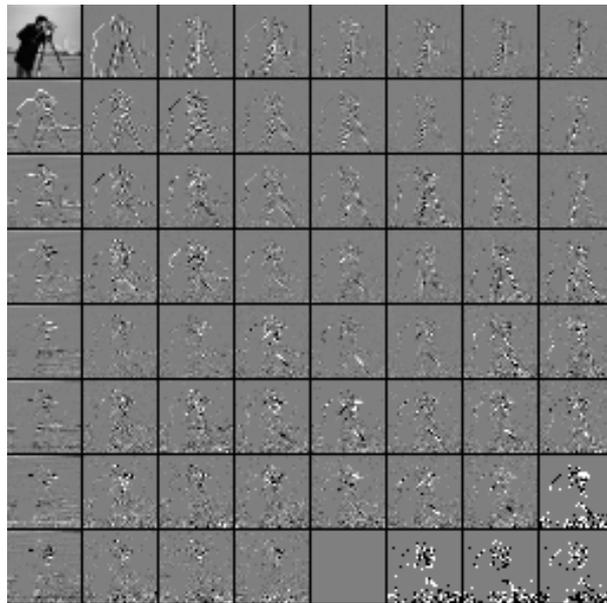
Transmission progressive d'image correspondant à une allocation de débit variable du codage DCT2d.

La norme Jpeg utilise en outre un codage entropique qui n'est pas pris en compte ci-dessous :

*originale*

*DCT2d quantifiée*

*DCT2d inverse*

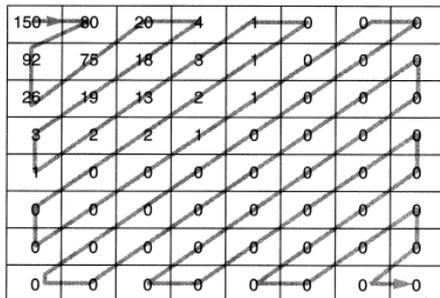


256x256 pixels codés  
sur 8 bits.

Bitrate : 4  
Size : 256 Ko  
Mean Square Error : 7,5

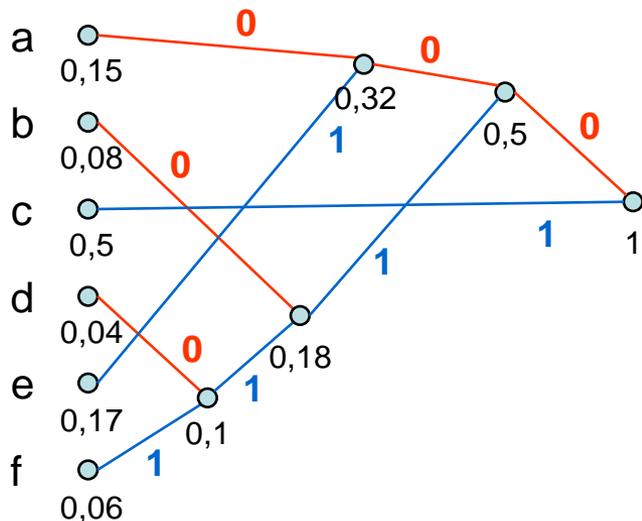
Après la transformée en DCT2d et la quantification, la deuxième partie de la norme Jpeg utilise une *compression sans perte* qui va : (1) regrouper les nombreuses valeurs nulles par plages (*Run Length Encoding*), et (2) exploiter le déséquilibre dans la distribution des valeurs à coder en attribuant à chaque valeur (symbole) une longueur (nombre de bits) inversement proportionnelle à sa probabilité d'apparition (*codage entropique*).

(1) Chaque bloc DCT2d quantifié est transformé en un vecteur ligne grâce à un parcours en zig-zag :



Cette transformation regroupe les valeurs faibles en fin de vecteur.

(2) Les valeurs non nulles sont codées par le codage de Huffman :



| Symbole | Probab. | Code |
|---------|---------|------|
| a       | 0,15    | 000  |
| b       | 0,08    | 010  |
| c       | 0,5     | 1    |
| d       | 0,04    | 0110 |
| e       | 0,17    | 001  |
| f       | 0,06    | 0111 |