

## Technologie Multimédia



### Chapitre 4: Introduction à la compression des images

#### 1. Généralités

- Une image numérique (ou discrète) est une image constituée d'un nombre fini de points.
- **Point = pixel (picture element)**
- **Définition : dimension de l'image en pixel**
- **Résolution : nombre de pixel sur une unité de surface (point par pouce)**
- **Profondeur : nombre de bit représentant un pixel de l'image**



3 → 1. Généralités

### Mode de stockage des images

**Bitmap** : c'est le mode de représentation le plus utilisé car il permet a priori de représenter tout type d'image

**Vectoriel** : l'image est stockée sous la forme d'une définition mathématique ( cercle = position du centre + rayon + ...)

**Taille d'un fichier image:**

La taille d'un fichier image non compressé est donnée par :  $T = DP$

D : définition (taille) de l'image, P : profondeur de l'image



4 → 2. Compression de données

**La compression de données** a pour but de réduire l'espace nécessaire à la représentation d'une certaine quantité d'information.

**Compression non destructive** : pas de perte d'information.  
On parle de **compactage**

**Compression destructive** : perte d'information

5 → **2. Compression de données**

**Evaluation de la compression et des pertes**

- Quotient de compression :  $Q = \frac{\text{Taille\_Initiale}}{\text{Taille\_Finale}}$
- Taux de compression :  $T = \frac{1}{Q}$
- Gain de compression :  

$$G = 1 - T = \frac{\text{Taille\_Initiale} - \text{Taille\_Finale}}{\text{Taille\_Initiale}}$$

FSG  
Faculté des Sciences de Géosciences

6 → **2. Compression de données**

**Evaluation de la compression et des pertes**

- EQM : Erreur Quadratique moyenne ([mean squared error](#))  
Entre deux images A et B de même taille MxN

$$EQM = \frac{1}{M * N} \sum_{i=1}^N \sum_{j=1}^M (A(i,j) - B(i,j))^2$$

FSG  
Faculté des Sciences de Géosciences

7 2. Compression de données

Qu'est ce qu'une bonne compression ?

Un algorithme performant de compression possède un gain de compression maximal et une erreur quadratique moyenne minimale

**G↑ & EQM↓**

**ATTENTION :** compte tenu des information effectivement perçues par notre œil, il est possible d'avoir à la fois une image de "qualité" et une EQM importante !



The slide features a dark blue header bar with the number 7 and the title '2. Compression de données'. Below the header, there is a question 'Qu'est ce qu'une bonne compression ?' followed by a statement about a good compression algorithm having maximum compression gain and minimum mean squared error. A large green upward arrow and a red downward arrow are positioned next to the text 'G↑ & EQM↓'. At the bottom, a warning in red text states 'ATTENTION : compte tenu des information effectivement perçues par notre œil, il est possible d'avoir à la fois une image de "qualité" et une EQM importante !'. The FSG logo is located in the top right corner.

8 3. Techniques de compression non-destructives

- RLE (Run-Length Encoding)
- Méthodes statistiques
  - Théorie de l'information
  - Codage de Huffman
  - Codage arithmétique
- Méthodes par dictionnaire



The slide features a dark blue header bar with the number 8 and the title '3. Techniques de compression non-destructives'. Below the header, there is a list of compression techniques. The first item is 'RLE (Run-Length Encoding)'. The second item is 'Méthodes statistiques', which includes three sub-points: 'Théorie de l'information', 'Codage de Huffman', and 'Codage arithmétique'. The third item is 'Méthodes par dictionnaire'. The FSG logo is located in the top right corner.

9 3. Techniques de compression non-destructives

RLE (Run-Length Encoding)

**Principe**

Si une donnée **d** apparaît **n** fois consécutivement dans le flux d'entrée, remplacer les n occurrences par la paire "**n d**".

Exemple :

**ABBBBBCAAAACCC** --> **1A 4B 1C 3A 3C** (sans les espaces)

Gain de compression :

**G = (12-10)/12 = 16,7%**



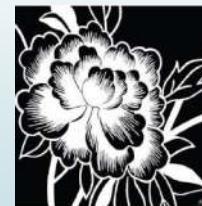
10 3. Techniques de compression non-destructives

RLE (Run-Length Encoding)

**Application aux images monochromes**

Deux couleurs possibles : N (0) & B (1)  
ex: 0, 0, 0, 1, 1, 1, 0, 0, 0 ...

On suppose que la couleur de départ est le noir. On compresse alors uniquement en indiquant le nombre de pixels N ou B successifs  
ex : Image originale : [0, 0, 0, 1, 1, 1, 0, 0, 0]  
Image compressée en RLE : [(3, 0), (3, 1), (3, 0)]



**3. Techniques de compression non-destructives**

RLE (Run-Length Encoding)

**Application aux images en niveau de gris**

ex : Image possédant une profondeur de 8 bits :

... 12, 12, 12, 12, 12, 35, 76, 112, 67, 87, 87, 87, 87 ...

codée en :

5 12 35 76 112 67 4 87 ou bien...

5 12 1 35 1 76 1 112 1 67 4 87

*Comment distinguer ces nombres ?*



FSG  
Faculté des Sciences de Géosciences

**12. Techniques de compression non-destructives**

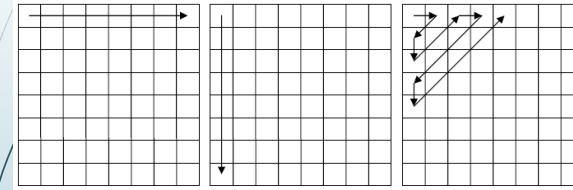
RLE (Run-Length Encoding)

**Comment parcourir l'image ?**

Codage suivant l'axe des X

Codage suivant l'axe des Y

Codage ZIG ZAG



FSG  
Faculté des Sciences de Géosciences

### 13. Techniques de compression non-destructives

#### RLE (Run-Length Encoding)

##### Application sur le format BMP (codage sur 8 bits, RGB)

- Les pixels compressés sont organisés par paire d'octets :

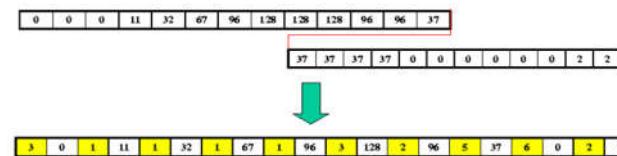
**04 02 = 02 02 02 02**

- La valeur **00** est un caractère d'échappement, et sa signification dépend de l'octet suivant :
  - 00 00** = eol (end of line)
  - 00 01** = eni (end of image)
  - 00 02** = saut dans l'image. Les 2 octets suivants indiquent le nombre de ligne et de colonne.
  - 00 XX** = XX raw pixels. Les XX octets suivants sont les valeurs des pixels.

### 14. Techniques de compression non-destructives

#### RLE (Run-Length Encoding)

##### Application sur le format BMP (codage sur 8 bits, RGB)



##### Avantage

- Algorithme de compression et décompression très simples et rapides.

##### Limites

- Efficace seulement pour nombreuses et longues plages constantes.
- Cas des images de synthèse simples ; peu adapté aux photos.
- Utilisé ponctuellement dans de nombreux formats (BMP, JPG, TIFF, PCX)

### 3. Techniques de compression non-destructives

15

**Méthodes statistiques**

**Le code de Huffman**

Le **principe** créer une structure d'arbre composée de nœuds. Chaque nœud de l'arbre représente un symbole, et le poids d'un nœud est égal à la fréquence de son symbole correspondant. Les nœuds feuilles de l'arbre représentent les symboles de la source, et les nœuds internes représentent des groupes de symboles. Les symboles les plus fréquents sont représentés par les nœuds les plus proches de la racine de l'arbre, tandis que les symboles les moins fréquents sont représentés par les nœuds les plus éloignés de la racine de l'arbre.

**Méthode** : construction d'un arbre (binaire, dans le cas d'un code binaire)

### 3. Techniques de compression non-destructives

Soit une source émettant les symboles  $u_1, \dots, u_6$ , avec des probabilités de 0.4, 0.3, 0.1, 0.1, 0.06 et 0.04 respectivement.

**Phase 1 : Construction de l'arbre.**

**Etape 1:** Classer les symboles par ordre décroissant de probabilité

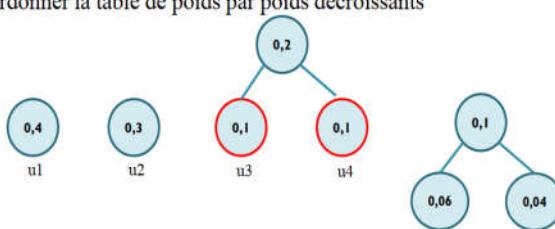
**Etape 2:** Réunir les 2 plus faibles probabilités en une seule "super-proba"



Faculté des Sciences de Gestion

## 13 Techniques de compression non-destructives

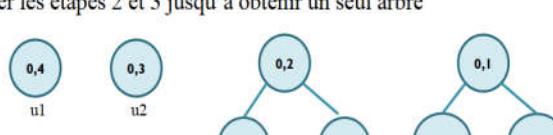
Etape 3: Réordonner la table de poids par poids décroissants



```

graph TD
    Root((0,2)) --- u1((0,4))
    Root --- u2((0,3))
    u1 --- u3((0,1))
    u2 --- u4((0,1))
    u3 --- u5((0,06))
    u4 --- u6((0,04))
  
```

Etape 4: répéter les étapes 2 et 3 jusqu'à obtenir un seul arbre



```

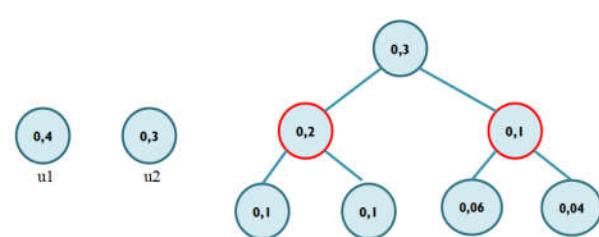
graph TD
    u1((0,4))
    u2((0,3))
    u3((0,2))
    u4((0,1))
    u5((0,06))
    u6((0,04))
  
```


  
 Institut für Geographie und Globalen Studien

18

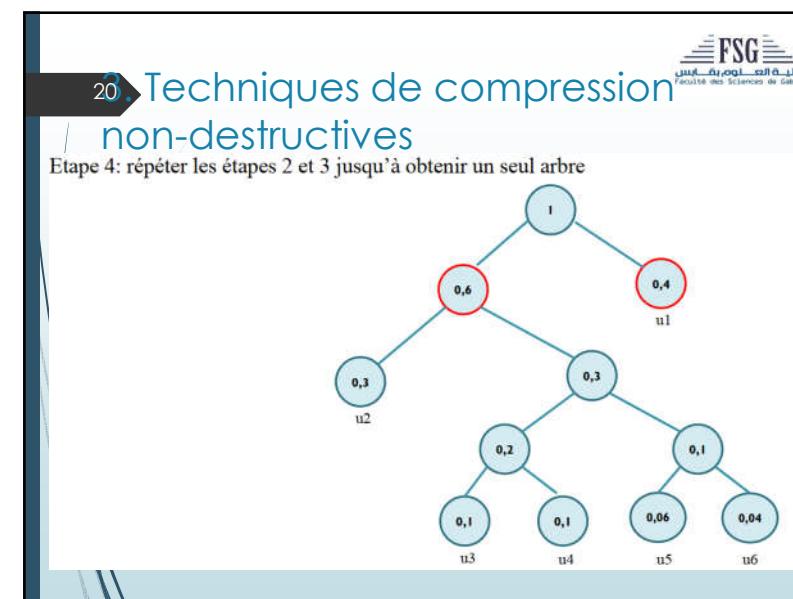
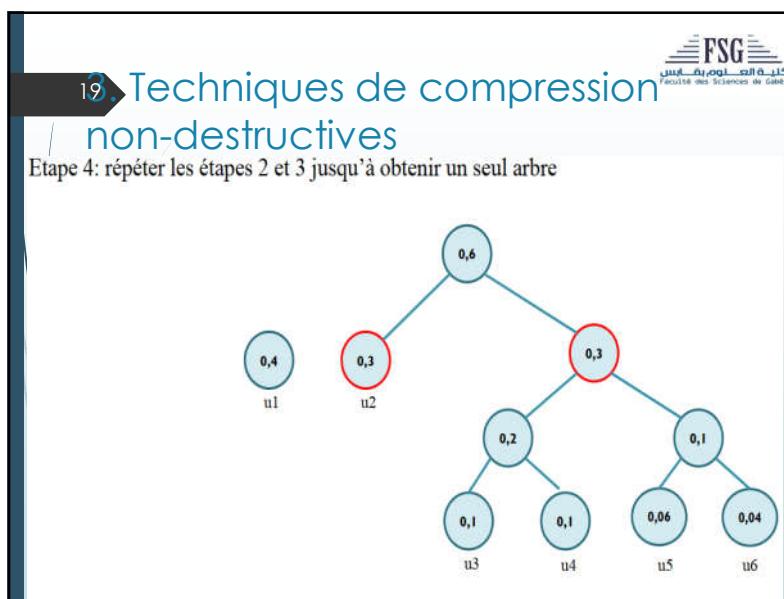
### 3. Techniques de compression non-destructives

Etape 4: répéter les étapes 2 et 3 jusqu'à obtenir un seul arbre



```

graph TD
    Root[0,3] --- N02((0,2))
    Root --- N01((0,1))
    N02 --- u3((0,1))
    N02 --- u4((0,1))
    N01 --- u5((0,06))
    N01 --- u6((0,04))
  
```



**23. Techniques de compression non-destructives**

Phase 2 : Construction du code à partir de l'arbre obtenu dans la phase 1.

```

graph TD
    Root((0.6)) -- 0 --> Node1((0.3))
    Root -- 1 --> Node2((0.3))
    Node1 -- 0 --> Node3((0.1))
    Node1 -- 1 --> Node4((0.1))
    Node2 -- 0 --> Node5((0.06))
    Node2 -- 1 --> Node6((0.04))
    Node3 -- 0 --> u2((0.3))
    Node3 -- 1 --> u3((0.1))
    Node4 -- 0 --> u4((0.1))
    Node4 -- 1 --> u5((0.06))
    Node5 -- 0 --> u6((0.04))
    Node5 -- 1 --> u1((0.4))
  
```

**23. Techniques de compression non-destructives**

Phase 2 : Construction du code à partir de l'arbre obtenu dans la phase 1.

Lecture du codage obtenu,  
du haut vers le bas

Symbol	Probabilité	Code
u1	0,4	"1"
u2	0,3	"00"
u3	0,1	"0100"
u4	0,1	"0101"
u5	0,06	"0110"
u6	0,04	"0111"

### 23. Techniques de compression non-destructives

Symbole	Probabilité	Nombre de bits (avant compression)	Nombre de bits (après compression)
u1	0,4	8	1
u2	0,3	8	2
u3	0,1	8	4
u4	0,1	8	4
u5	0,06	8	4
u6	0,04	8	4

Poids avant compression : 48 bits;

Poids après compression : 19 bits

### 24. Techniques de compression non-destructives

#### Méthodes statistiques

##### Le code de Huffman

##### Conclusion

- Technique de codage très souvent utilisée comme "surcouche" à d'autres méthodes de compression (ex: images TIFF, JPEG, ou MPEG, ...)
- Mais ... elle nécessite la connaissance préalable des probabilités d'apparition des symboles de source.

## 3. Techniques de compression non-destructives

**Méthodes Par dictionnaire**

Les algorithmes de type dictionnaire établissent un **dictionnaire** des termes employés dans un fichier, termes qui peuvent être de taille quelconques, et on leur associent un code de taille moins importante.

Le fichier compacté est composé du dictionnaire et du fichier où les termes sont remplacés par les codes.

Codage  $\rightarrow C = AAAAAA$

## 3. Techniques de compression non-destructives

**Méthodes Par dictionnaire**

Ces méthodes ont été introduites par **A. Lempel** et **J. Ziv** dans les années 70. On parle aujourd'hui couramment de **méthodes LZ**.

L'idée principale est de remplacer le symbole à coder par son **index (code)** qui le représente dans le dictionnaire.

Ce dictionnaire peut être **statique et connu à l'avance, ou dynamique, construit au fur à et à mesure de la compression**.

Le contenu de ce dictionnaire conditionne complètement les performances de la compression.

26

## 3. Techniques de compression non-destructives

### Méthodes Par dictionnaire

Cette méthode utilise deux passes. Chaque passe comporte trois étapes. La figure détaille chaque étape.

Méthode de compression en 2 passes	
1 <sup>ère</sup> passe	2 <sup>ième</sup> passe
Construction d'une liste contenant l'ensemble des symboles ainsi que leur fréquence d'apparition.	Relecture du fichier pour en effectuer la compression
Classement de la liste par ordre décroissant des fréquences	Chaque octet est remplacé par son index dans la liste, codé selon 1 à 8 bits
La liste classée devient le dictionnaire. Elle est écrite dans la sortie du compresseur.	Ajouter 3 bits indiquant la taille de cette index

**Figure 4.2** Etapes de compression par approche dictionnaire statique

## 3. Techniques de compression non-destructives

### Méthodes Par dictionnaire

**Exemple de compression**

Appliquer la compression par dictionnaire statique à la chaîne « CITRONTRESCONTRIT »

Première passe : construction du dictionnaire

Etape 1 : liste de symbole avec fréquence		Etape 2 : classement de symbole selon fréquence		Etape 3 : dictionnaire		
Car	freq	Car	freq	Car	freq	Code
C	2	T	4	T	4	0
I	2	R	3	R	3	1
T	4	C	2	C	2	00
R	3	I	2	I	2	01
O	2	O	2	O	2	10
N	2	N	2	N	2	11
E	1	E	1	E	1	000
S	1	s	1	s	1	001

### 3. Techniques de compression non-destructives

**Méthodes Par dictionnaire**

Deuxième passe : construction du code

Etape 1: Lecture symbole par symbole	Etape 2: remplacement du symbole par son code	Etape 3: ajout de la longueur du code
C I T R O N T ...	00 01 0 ....	00100001010000 ....

Etape 3 : dictionnaire

Car	freq	Code
T	4	0
R	3	1
C	2	00
I	2	01
O	2	10
N	2	11
E	1	000
s	1	001

### 3. Techniques de compression non-destructives

**Méthodes Par dictionnaire: LZ**

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

0000000100111010111000100110

Essayer de décompresser cette suite de valeur sachant que chaque index (code) est précédé de 3 bits indiquant la taille de cet index

Dictionnaire :

code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

30

**3. Techniques de compression non-destructives**

**Méthodes Par dictionnaire**

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

0000000100111010111000100110

**Dictionnaire :**  
Longueur de l'index = 1

code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

31

**3. Techniques de compression non-destructives**

**Méthodes Par dictionnaire**

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

0000000100111010111000100110

code de l'index = 0

Image :  
132

code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

32

**3. Techniques de compression non destructives**

Méthodes Par dictionnaire

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

0000000010011010111000100110

**Dictionnaire :**  
Longueur de l'index = 1  
code de l'index = 1

**Image :**  
132 177

**Dictionnaire :**

code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

33

**3. Techniques de compression non destructives**

Méthodes Par dictionnaire

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

0000000010011010111000100110

**Dictionnaire :**  
Longueur de l'index = 2  
code de l'index = 11

**Image :**  
132 177 72

**Dictionnaire :**

code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

34

## 3. Techniques de compression non-destructives

**Méthodes Par dictionnaire**

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

000000010011101011000100110

**Dictionnaire :**  
Longueur de l'index = 3  
code de l'index = 111

**Image :**  
132 177 72 59

code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

35

## 3. Techniques de compression non-destructives

**Méthodes Par dictionnaire**

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

0000000100111010111000100110

**Dictionnaire :**  
Longueur de l'index = 1  
code de l'index = 1

**Image :**  
132 177 72 59 17711

code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

36

### 3. Techniques de compression non-destructives

#### Méthodes Par dictionnaire

**Exemple:** séquence de pixels en niveau de gris compressés par LZ ainsi que le dictionnaire employé.

000000010011101011100010 10

**Dictionnaire :**  
Longueur de l'index = 2

code de l'index = 10

**Image :**  
132 177 72 59 177 35



Dictionnaire :	
code	ndg
"0"	132
"1"	177
"11"	72
"10"	35
...	...
"111"	59

37

### 3. Techniques de compression non-destructives

Code	Intensité
0	132
1	177
11	72
10	35
111	59

Le schéma de décompression est :

Code	000	0	000	1	001	11	010	11	000	1	001	10
Index	1		1		2		3		1		2	
Message	132		177		72		59		177		35	

## 4. Techniques de compression destructives

**Généralité**

- Ces techniques de compression sont avec pertes d'information.
- Ces méthodes permettent de retrouver une **approximation** de l'image numérique.
- Les pertes sont **généralement** indétectables à l'œil nu
- Les méthodes de compression les plus souvent utilisées sont : la norme JPEG (Joint Photographic Expert Group), la compression par ondelettes, Transformation en Cosinus Discrète (TCD) , l'utilisation des fractales,...

39

## 4. Techniques de compression destructives

**Principe fondamental**

Etant donné un pixel, il y a de forte chance que ses voisins possèdent la même couleur, ou du moins une couleur similaire

Les techniques de compression avec perte sont donc basées sur le fait que les niveaux de gris de pixels voisins sont **fortement corrélés (reliés)**.

On parle alors de **redondance (répétition) spatiale**.

C'est cette corrélation qui permet la compression.

40



## 4. Techniques de compression destructives

### Premières méthodes intuitives

**Sous échantillonnage** : on ignore simplement certains pixels. Les effets sur l'image sont très visibles (grande perte de détails) : cette méthode simple est très peu utilisée.

**Sous échantillonnage des couleurs** : notre œil est moins sensible aux variations de chrominance que de luminance. Les deux chrominances sont sous-échantillonnées d'un facteur 2, conduisant à une réduction de 50% de la taille du fichier

41



## 3. Techniques de compression destructives

- **JPEG (Joint Photographic Experts Group)**

Le JPEG (Joint Photographic Experts Group) n'est pas un nom de fichier, mais **une méthode de compression**.

C'est aujourd'hui une des méthodes de compression les plus utilisées, notamment grâce aux forts taux de compression qu'elle est capable d'atteindre

42

**3. Techniques de compression destructives**

**JPEG (Joint Photographic Experts Group): mesure de la distortion**

**EQM**

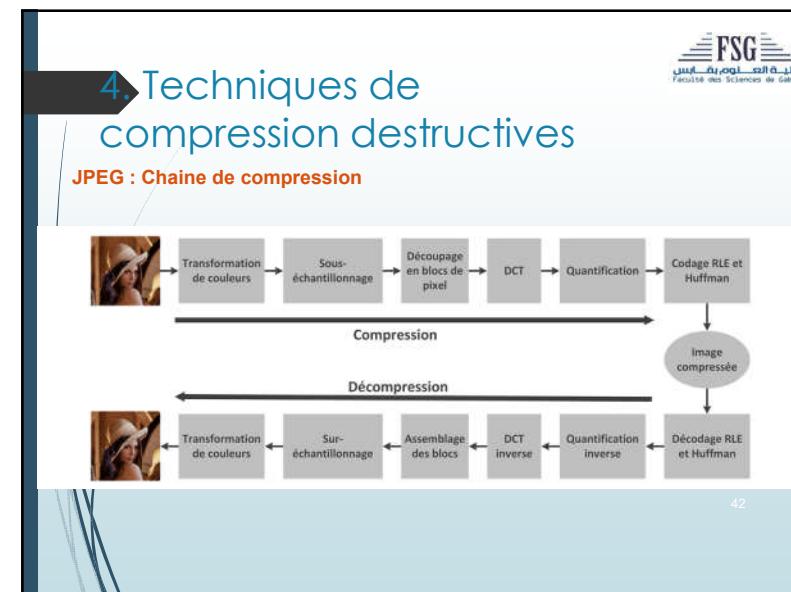
- On calcule l'erreur quadratique moyenne (EQM) entre l'image originale et celle reconstruite après la compression.

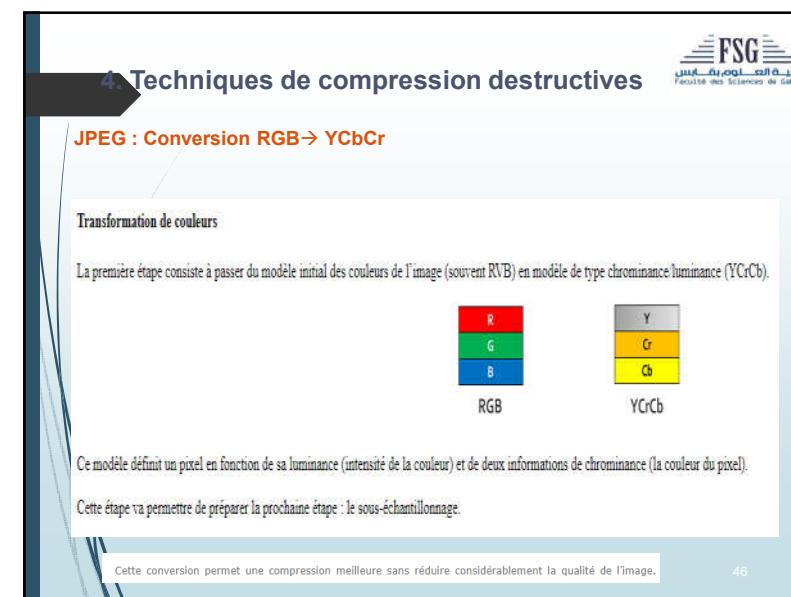
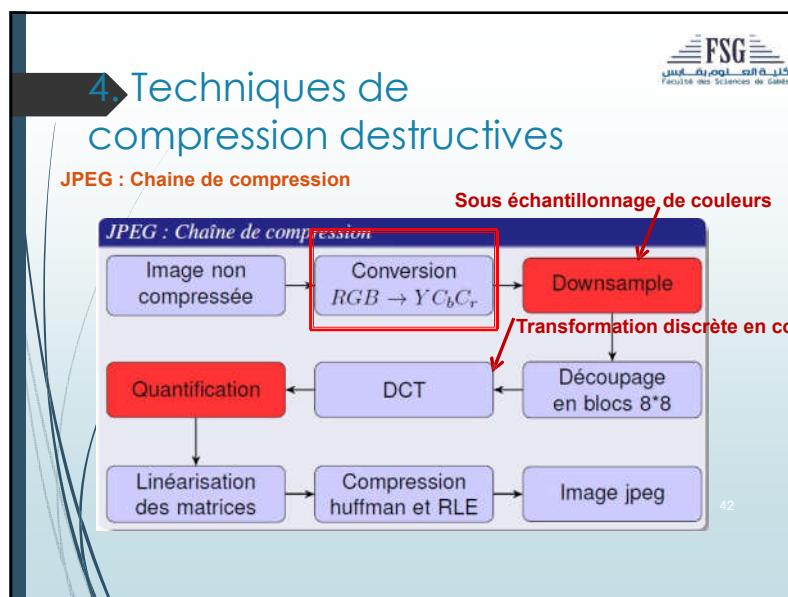
$$EQM = \frac{1}{M * N} \sum_{i=1}^N \sum_{j=1}^M (A(i,j) - B(i,j))^2$$

- Tel que M et N représentent les dimensions de l'image.  
A : Image avant compression  
B : Image après décompression

La reconstitution de l'image est plus performante quand cette mesure est plus petite.

43





**4. Techniques de compression destructives**

**JPEG : Conversion RGB → YCbCr**

**Conversion  $RGB \rightarrow YC_bC_r$**

- Pour calculer les valeurs des composantes YCbCr d'une image à partir des composantes RGB/RVB (qui varient de 0 à 255, on utilise les formules suivantes :
 
$$Y = 0,299 * R + 0,587 * G + 0,114 * B$$

$$C_b = -0,1687 * R - 0,3313 * G + 0,5 * B + 128$$

$$C_r = 0,5 * R - 0,4187 * G - 0,0813 * B + 128$$
 L'ajout de 128 à  $C_b$  et  $C_r$  permet d'obtenir des valeurs entre 0 et 255
- La conversion inverse se fait ainsi (les valeurs obtenues varient encore entre 0 et 255) :
 
$$R = Y + 1,402 * (C_r - 128)$$

$$G = Y - 0,34414 * (C_b - 128) - 0,71414 * (C_r - 128)$$

$$B = Y + 1,772 * (C_b - 128)$$

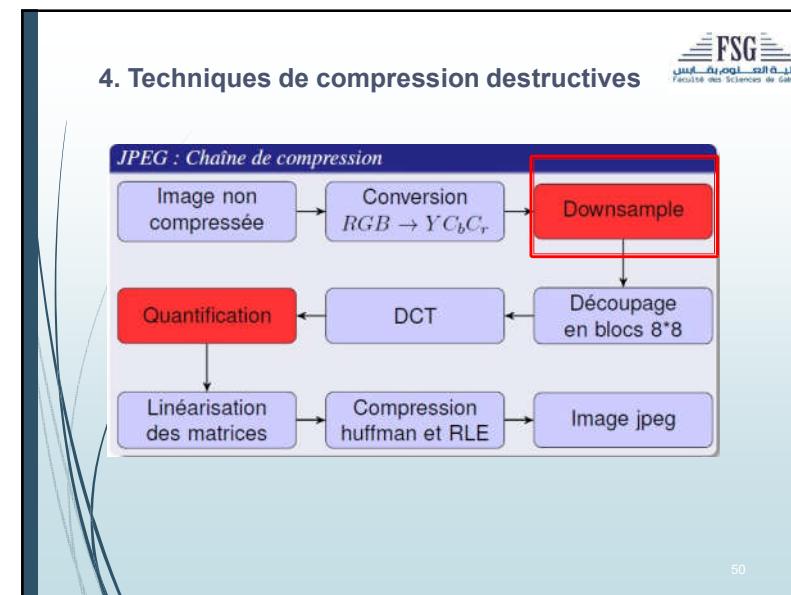
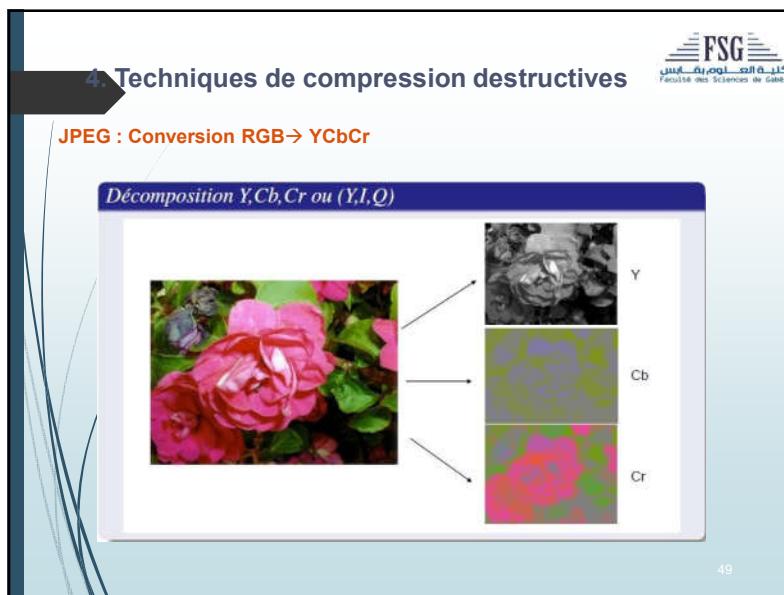
47

**4. Techniques de compression destructives**

**JPEG : Conversion RGB → YCbCr**

**Décomposition R,G,B**

48



## 4. Techniques de compression destructives



### Le sous échantillonnage:

- Le sous-échantillonnage est un processus qui consiste à réduire la quantité de données d'une image.
- La sensibilité de l'œil humain est beaucoup plus importante à la composante luminance qu' aux composantes chrominance .

51

## 4. Techniques de compression destructives



### JPEG : Downsample

Le 4:4:4 est un sous-échantillonnage qui consiste à ne rien faire. Aucune compression n'est effectuée et aucune perte de qualité ne peut être enregistrée.

Y	Y	Y	Y	Y	Y	Y	Y	Y
Cb								
Cr								
Y	Y	Y	Y	Y	Y	Y	Y	Y
Cb								
Cr								
Y	Y	Y	Y	Y	Y	Y	Y	Y
Cb								
Cr								
Y	Y	Y	Y	Y	Y	Y	Y	Y
Cb								
Cr								
Y	Y	Y	Y	Y	Y	Y	Y	Y
Cb								
Cr								
Y	Y	Y	Y	Y	Y	Y	Y	Y
Cb								
Cr								

52

## 4. Techniques de compression destructives

### JPEG : Downsample

Le 4:2:2 est un sous-échantillonnage qui consiste à supprimer les informations de la chrominance sur une colonne sur deux. La colonne restante possède alors la moyenne des chrominances des deux colonnes. Lors de la décompression, il suffira de recopier cette information sur la colonne vide.

$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$

53

## 4. Techniques de compression destructives

### JPEG : Downsample

Le 4:2:0 est un sous-échantillonnage qui effectue le même processus que le 4:2:2 mais en effectuant aussi cette opération sur les lignes. On divise donc ici par 4 l'information de la chrominance.

$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$
$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$	$Y_{Cb}$	$Y_{Cr}$

54

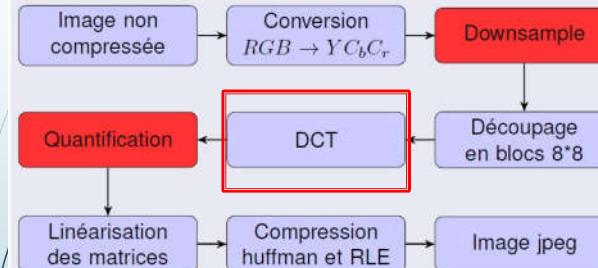
## 4. Techniques de compression destructives

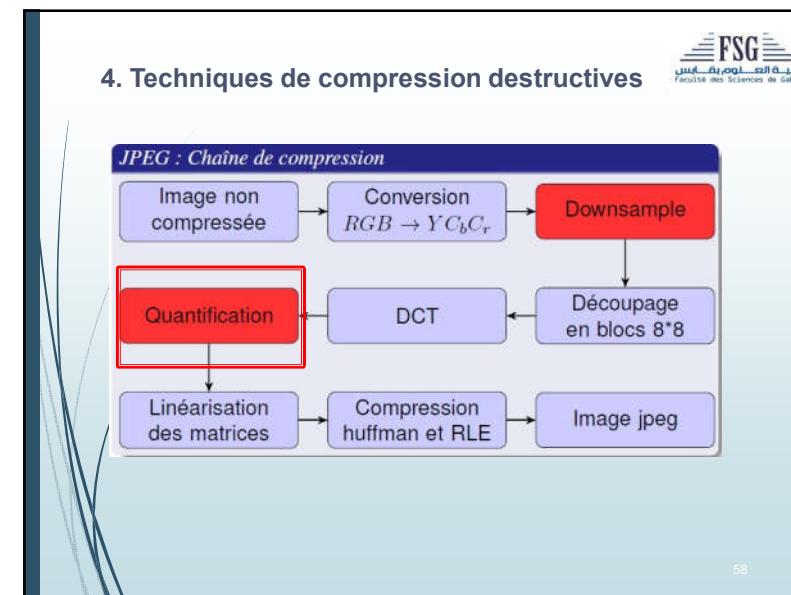
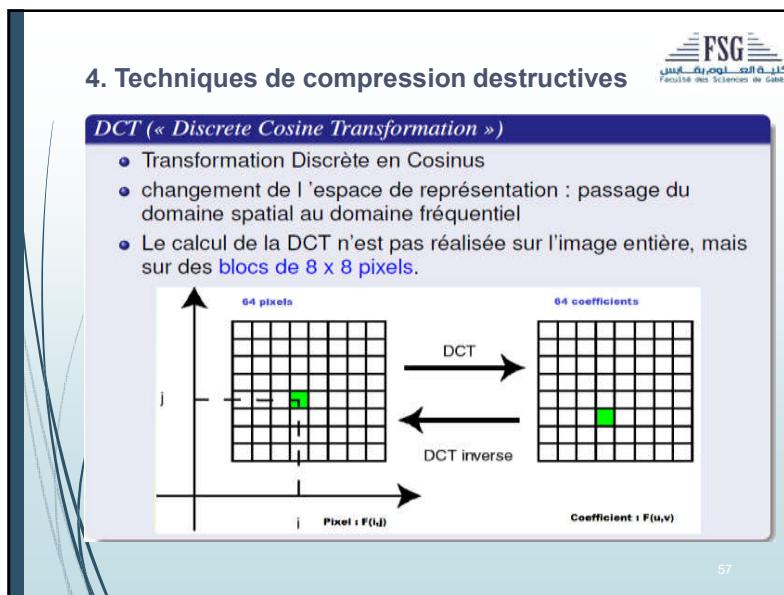
### JPEG : Downsample

- Le sous-échantillonnage de la chrominance est une méthode de réduction de volume des images numériques. Il consiste à diminuer le nombre d'échantillons à traiter : on parle de décimation
- La vision humaine présentant une sensibilité moindre à la couleur qu'à la luminosité, on conserve généralement moins d'informations de chrominance que de luminance sans pour autant dégrader la qualité perçue de l'image

## 4. Techniques de compression destructives

### JPEG : Chaîne de compression





**4. Techniques de compression destructives**

**JPEG : Quantification**

C'est dans l'étape de la quantification où on perd physiquement des informations et donc détérioration de la qualité de l'image mais on gagne en contrepartie de la place (ce qui est l'objet principal de la compression).

On a à notre disposition les matrices retournées par la DCT, pour réduire les hautes fréquences (pour perdre les informations dont l'œil n'est pas très sensible c'est-à-dire la chrominance et ainsi gagner de l'espace) on divise ces matrices sur une matrice appelée la **table de quantification ou matrice de quantification** (diviser termes à termes les valeurs  $C(i)$  et  $C(j)$  par des valeurs).

L'objectif principal de la quantification est de coder les coefficients sur un minimum de valeurs possibles, c'est-à-dire moins de bits pour un coefficient donné.

59

**4. Techniques de compression destructives**

**JPEG : Quantification**

**Exemple :**

Appliquons le principe de la quantification sur le coefficient 51 avec coefficient de quantification = 10 :

51 (décimal) > 110011 (binnaire) > 6 bits.  
 Arrondir (51/10)=arrondir (5,1)=5 (décimal) > 101 (binnaire) > 3 bits.

$5 \times 10 = 50 \sim 51$  (une petite erreur de quantification > perte d'informations).

Cet exemple montre comment la quantification réduit le nombre de bits et met en évidence la perte d'informations.

60

**4. Techniques de compression destructives**

### JPEG : Quantification

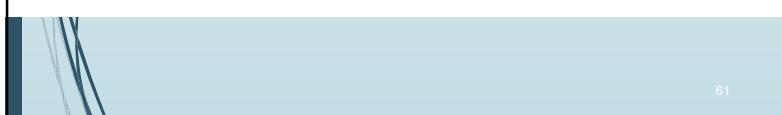
La table de quantification est paramétrable, c'est-à-dire qu'en changeant un paramètre on peut obtenir différents niveaux de compression. Ce paramètre est nommé le **facteur de qualité**, relié au niveau de compression souhaité. Le résultat de la quantification n'étant pas toujours entier, il s'agit d'arrondir les valeurs obtenues, vers les valeurs entières les plus proches.

$$Q(i,j) = 1 + (1+i+j)F_q$$

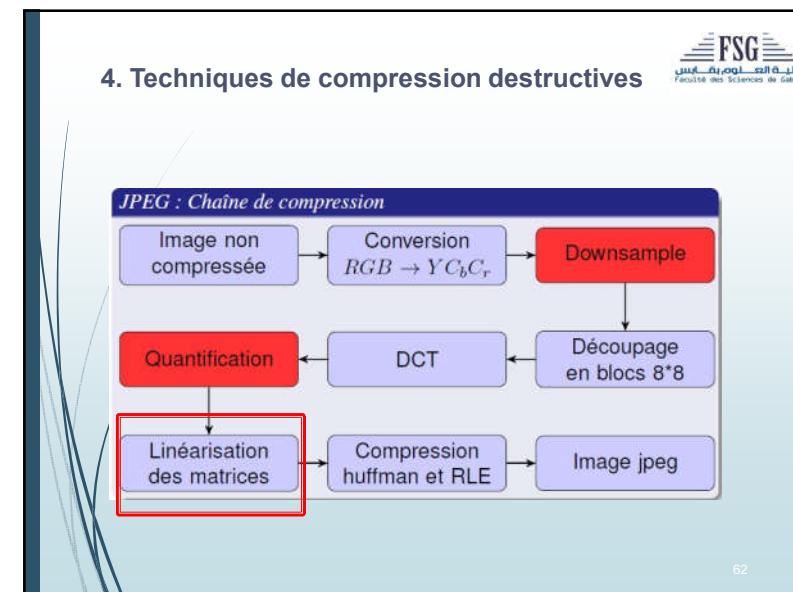
- $Q(i,j)$  : valeur du pas de quantification dans un bloc ( $i = 0 \dots N - 1$  et  $j = 0 \dots N - 1$ ).
- $F_q$  : facteur de qualité de la quantification.

Pour obtenir la matrice quantifiée on divise la matrice traitée par la DCT  $F$  sur la matrice de quantification  $Q$  :

Matrice quantifiée =  $(F(u,v) / Q(u,v))$



61



**4. Techniques de compression destructives**

**JPEG : linéarisation de l'image**

*Lecture en zigzag*

63

**4. Techniques de compression destructives**

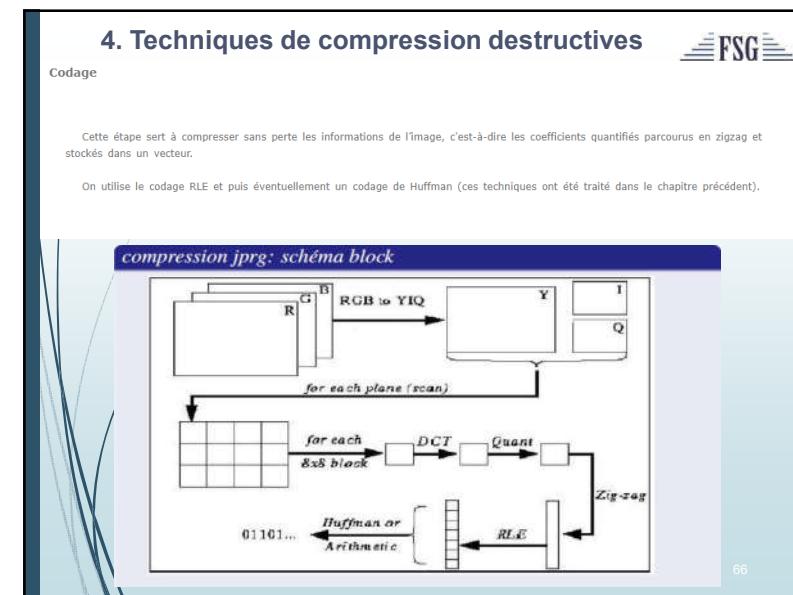
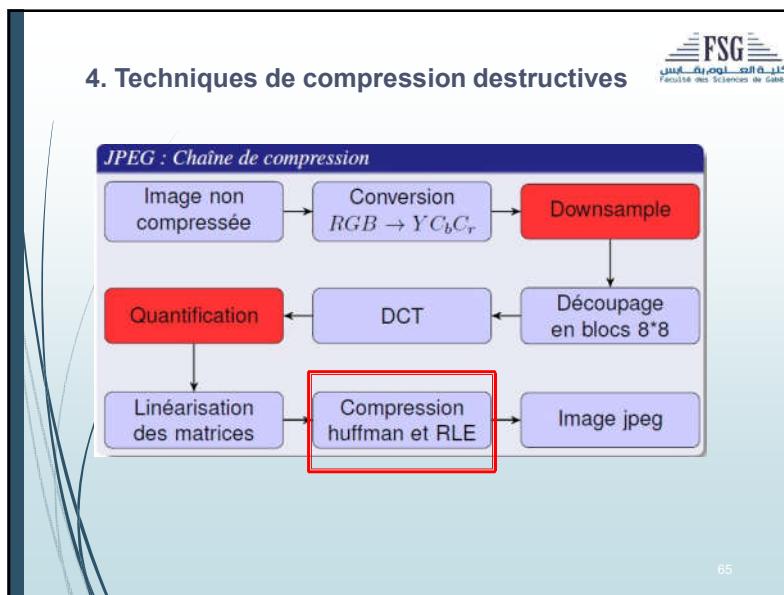
**JPEG : linéarisation de l'image**

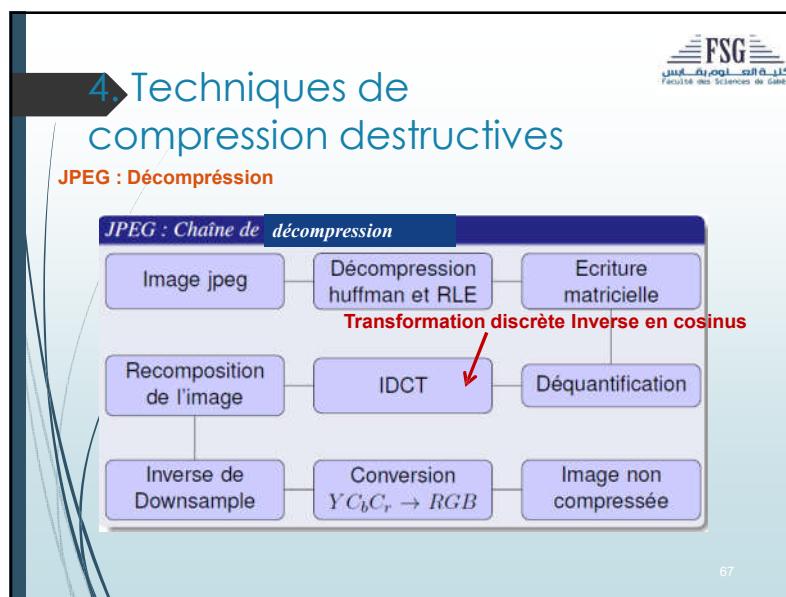
**Zigzag :**

On veut maintenant former un vecteur où les coefficients relatifs aux basses fréquences (haut à gauche) sont regroupés. Pour ce faire on a besoin du parcours en zigzag:

Ce parcours particulier permet de parcourir la matrice quantifiée de façon à former une sorte de dégradé de coefficients relatifs aux fréquences puisqu'en haut à gauche on a les basses fréquences et plus on va à droite et en bas on progresse vers les hautes fréquences, un parcours normal (vertical ou horizontal) est incapable de faire cela.

Ce parcours est aussi important parce que la quantification réduit plusieurs coefficients relatifs aux hautes fréquences au coefficient 0, on se retrouve donc avec une suite de zéros en fin du vecteur, il suffit donc d'appliquer un codage de type RLE par exemple pour réduire le nombre d'éléments du vecteur.



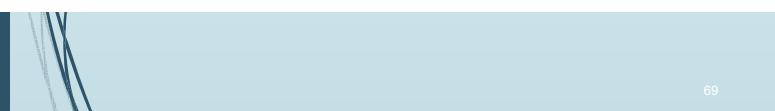




4. Techniques de

En bref

- Compression réalisée en **supprimant les hautes fréquences** (détails et contours).
  - ✓ **Perte** d'information peu visible.
  - ✓ **Bonne** compression.
- Formats d'image utilisant cette compression
  - ✓ **JPEG**.



69