Université Moulay Ismaïl
École Supérieure de Technologie de Meknès

Compression & Stockage

A.U: 2019/2020

Filière TSI: S4

Pr. Mostafa MERRAS

Version 1.2

Objectifs du cours

- Comprendre l'intérêt de compresser les données informatiques.
- Connaître différents types de compression.
- Introduire les principaux algorithmes de compression .

PLAN

- Introduction
- Compression sans perte (codage)
- Compression avec perte
- Compression audio/vidéo
- Conclusion

Introduction

- Position du problème
- Définition & Intérêts de la compression
- Types de compression
- Evaluation de la compression et des pertes

Introduction

Position du problème

→ Exemple 1: Transmission par télécopie

Une page à transmettre est constituée de points blancs et noirs. Chaque point est représenté par un élément binaire (“0” si la couleur est noire, “1” si elle est blanche). La taille (Largeur * Hauteur) de cette page est de **8,5x11 pouces.** Sachant que la résolution est de **200 points par pouce.**

1. Calculer le nombre d’éléments binaires nécessaires pour représenter cette page .

   (8,5 x 200) x (11 x 200) = 3,74 Mbits

2. Si on utilise un modem au débit de 14,4 Kbits/s, donner le temps nécessaire à la transmission de cette page.

   $3{,}74.10^6/14{,}4.10^3$ = **4 mins 20 s.**

Grâce aux méthodes de compression, cette durée est réduite à 17s !

Introduction

Position du problème

→Exemple 2: Fichiers musicaux

Considérons un signal analogique stéréo. Sa numérisation en "qualité CD" requiert une **fréquence d'échantillonnage de 44,1 KHz,** et une quantification des échantillons sur 16 bits (2 octets).

- Donnez le nombre d'élément binaire nécessaire à la représentation d'une **seconde** de musique (stéréophonique ⟺ Nombre de voies=2)

$44{,}1.10^3 \times 2 \times 2 = 176{,}4$ ko/s

- Avec un tél débit, donnez la durée qu'un CD de 650 Mo peut stocker .

650/0,1764 = 61 minutes de musique.

Grâce aux méthodes de compression, on peut stocker sur ce même CD plus de 10 heures de musique !

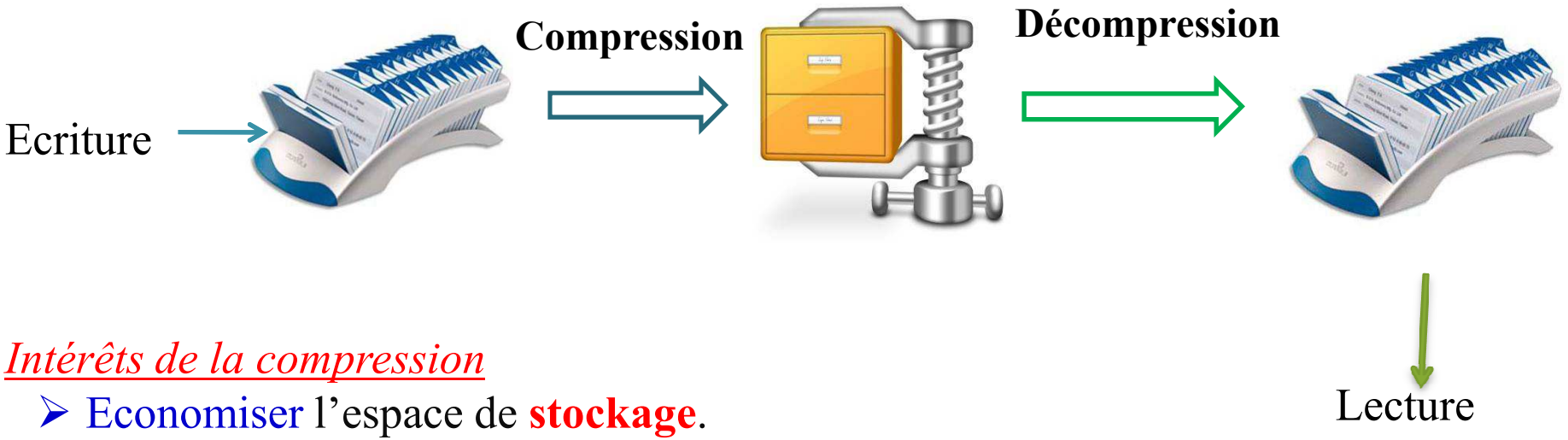
Introduction

Définition & Intérêts de la compression

Définition

- La compression de données informatiques consiste à **réduire** la taille de l'information pour son **stockage** et son **transport**.

- Le coût et les limites technologiques nécessitent d'utiliser la compression de données pour le stockage d'importants volumes d'information.



Intérêts de la compression

- Economiser l'espace de **stockage**.
- Diminuer le temps de **transmission** du fichier.

Introduction

Principe & Types de de la compression

Principe

1. Détection de **redondances** dans le signal
2. Un algorithme de *compression permet le codage **réduit** du signal*
3. Un algorithme (inverse) de *décompression permet **d'exploiter** le signal*

Types de compressions

❖**Compression sans perte (ou non-destructive, *i.e. codage ou compactage):***

➢Le signal obtenu après décompression est strictement **identique** à **l'original**

➢Utilisation : fichier exécutable, fichier texte

❖**Compression avec perte (ou destructive, ou *avec dégradation) :***

➢Le signal obtenu après décompression est **diffèrent** (légèrement) de **l'original**

➢Utilisation : image, son, vidéo

Introduction

Evaluation de la compression et des pertes

→**Quotient de compression** :

$$Q = \frac{\text{Taille\_Initiale}}{\text{Taille\_Finale}}$$

➢Plus une compression est **forte**, plus le **quotient** de compression est **élevé**.

→**Taux de compression**:

$$T = \frac{1}{Q}$$

→**Gain de compression (%)** :

$$G=(1-T)*100\%$$

➢Plus le **gain** de compression est **élevé**, plus la **taille** du fichier compressé résultant est **faible**.

→**Erreur quadratique moyenne** :

$$EQM = \frac{1}{M \times N} \times \sum_{n=1}^{N} \sum_{m=1}^{M} \left(X_{origine}(n,m) - X_{reconstruite}(n,m) \right)^2$$

Avec:

$X_{origine}$ (n,m) et $X_{reconstruite}$ (n,m) : représentent les valeurs des pixels d'image d'origine et reconstruite

Introduction

Evaluation de la compression et des pertes

- **Qu'est ce qu'une bonne compression ?**

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

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

Introduction

- Compression des principaux formats d'images bitmaps

Format	Espaces couleur	Compression(s)	C. α	Domaines d'utilisation, rem.
TIFF (.tif)	*RGB*, CIE $L^*a^*b^*$, *CMYB*, couleurs indexées, ndg	Aucune Sans perte (LZW, Huffman) Avec perte (JPEG)	Oui	PAO, Infographie, bureautique Très flexible, mais nombreuses variantes pas toujours supportées
BMP (.bmp)	*RGB*, couleurs indexées, ndg	Aucune Sans perte (RLE)	Non	Bureautique sous Windows Compression peu efficace
GIF (.gif)	couleurs indexées (2 à 256)	Sans perte (LZW)	Oui	Pages web Animations possibles
JFIF (.jpg)	*RGB*, *CMYB*, ndg	Avec perte (JPEG)	Non	Pages web, photographie Compr. efficace mais destructive
PNG (.png)	*RGB*, ndg, 256 couleurs indexées	Sans perte (deflate=LZ77+Huffman)	Oui	Pages web, photo, sans perte Format libre. Jusqu'à 48 bits.

Canal α: une 4éme couche qui vient s'ajouter aux 3 couches Rouge, Vert et Bleu (RVB), qui **permet** de stocker la **transparence** de l'image

PLAN

- Introduction
- Compression sans perte (codage)
- Compression avec perte
- Compression audio/vidéo
- Conclusion

Compression sans perte (codage)

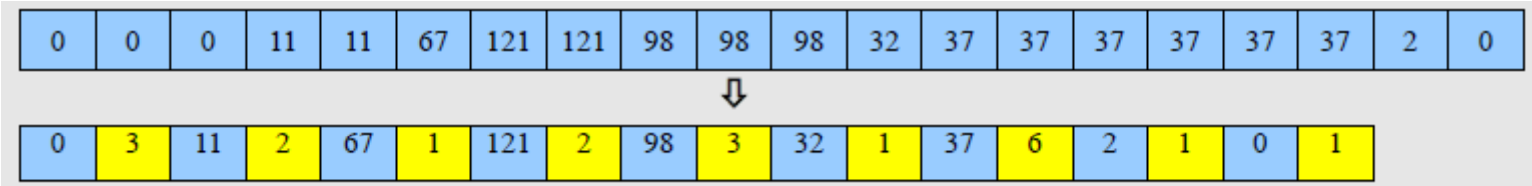
- RLE (Run-Length Encoding)
- Codage de Huffman
- Méthodes par dictionnaire

Compression sans perte (codage)

RLE (Run-Length Encoding)

Principe

- Codage par plage (« Running Length Encoding »).
- **Recherche** de séquences de données **redondantes** (ex. niveaux identiques).
- **Codage** de la **valeur** et du **nombre de répétitions**:



Avantage

- Algorithmes de compression et décompression très **simples** et **rapides**.

Limites

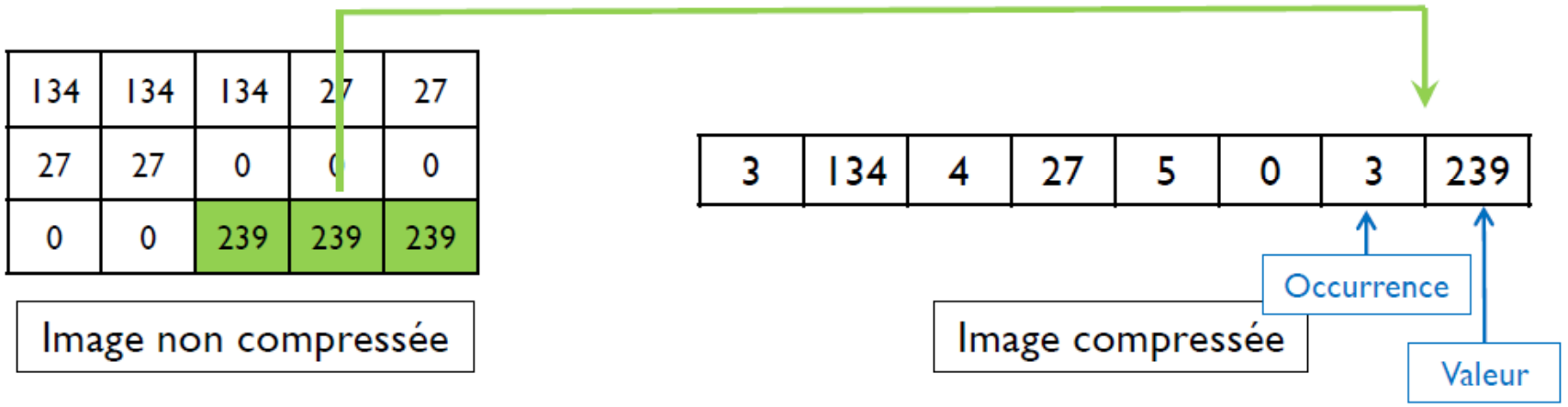
- Efficace seulement pour nombreuses et longues plages constantes.
  - Cas des images de synthèse simples ; peu adapté aux photos.
  - Utilisé *ponctuellement dans de nombreux formats (BMP, JPG, TIFF, PCX, ...)*.
- Nécessite de fixer un maximum pour la longueur des plages (ex. 255).

Compression sans perte (codage)

RLE (Run-Length Encoding)

Exemple 1: soit l'image en niveaux de gris ci-dessous (image non compressée)

Compression par RLE



Taille initale : 5 × 3 × 1(octet) = 15 *octet*

Taille de l'image compressée : 8 × 1 = 8 octet

Compression sans perte (codage)

RLE (Run-Length Encoding)

Exemple 2: Estimation du quotient de compression

Soit une chaîne de caractère de taille N à compresser. Supposons que cette chaîne contienne M répétions de longueur moyenne L d'un caractère. Quel est le quotient de compression si cette chaîne est compressée selon l'algorithme RLE ?

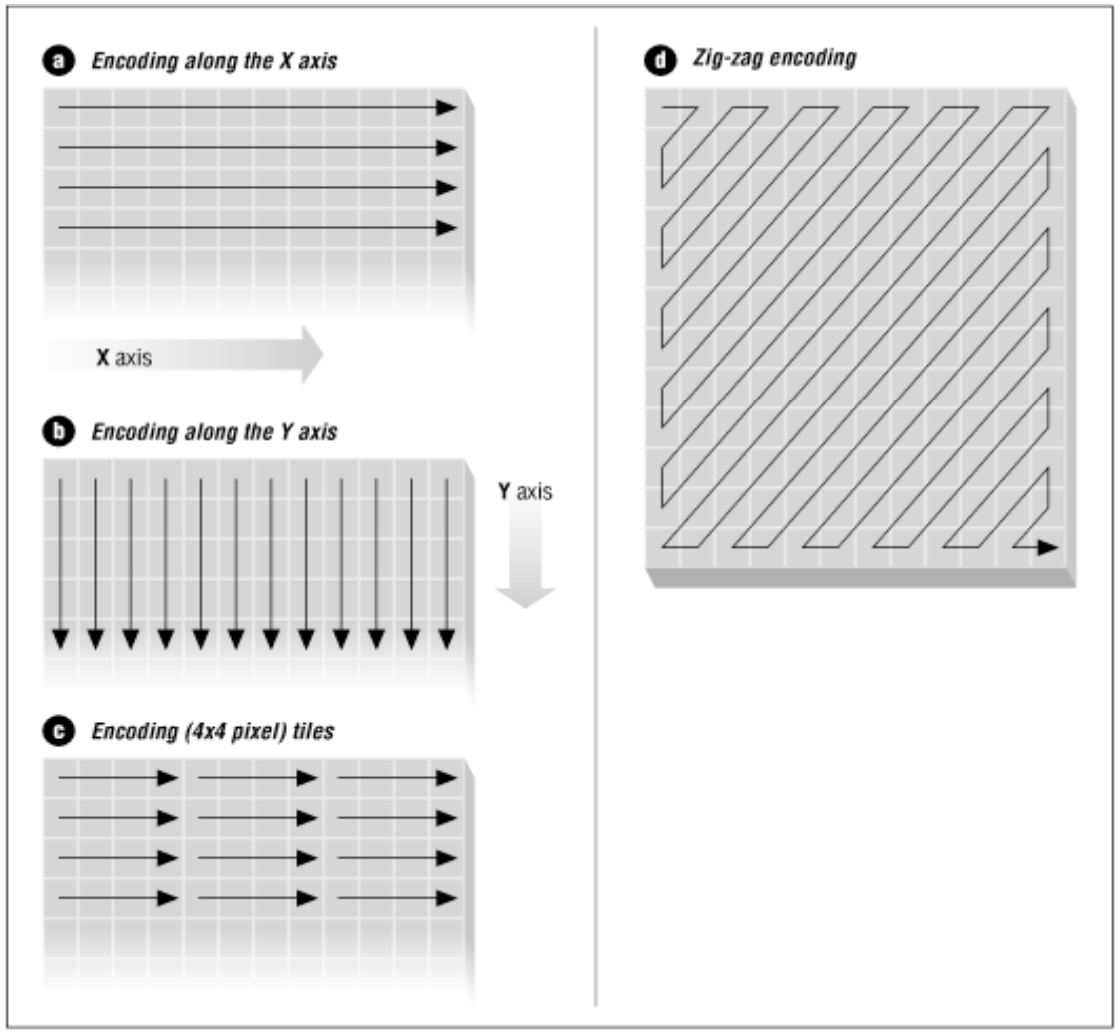
- ✓ Chacune des M répétitions est remplacée par 2 caractères
- ✓ Donc la taille de la chaîne compressée est N-ML+2M
- ✓ D'où

$$Q \approx \frac{N}{N - M(L-2)}$$

Compression sans perte (codage)

RLE (Run-Length Encoding)

Comment parcourir l'image ?



Compression sans perte (codage)

- RLE (Run-Length Encoding)
- Codage de Huffman
- Méthodes par dictionnaire

Compression sans perte (codage)

Codage de Huffman

Principe

- **Coder** les valeurs avec un **nombre de bits différent**.

  Code (utilisant des mots) à **longueur variable** (*ang. « Variable Length Coding »),* dit aussi codage entropique (*ang. « Entropy coding »).*

- Plus une **valeur apparaît fréquemment**, plus le **nombre** de **bits** utilisés pour la coder est **petit** (i.e. plus son code est court).

Algorithme de Huffman : codage

Phase 1 : Construction de l'arbre.

1. **Trier** les différentes valeurs par ordre **décroissant** de **fréquence d'apparition**
   - Table de ***poids.***
2. Fusionner les **deux poids minimaux** dans un arbre binaire et **affecter leur somme à la racine**.
3. **Réordonner** la table de poids par poids décroissants.
4. **Recommencer** en **2**. jusqu'à obtenir un **seul arbre**.

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

À partir de la racine, attribuer des **0** aux **sous-arbres de gauche** et des **1** à **droite**.

Compression sans perte (codage)

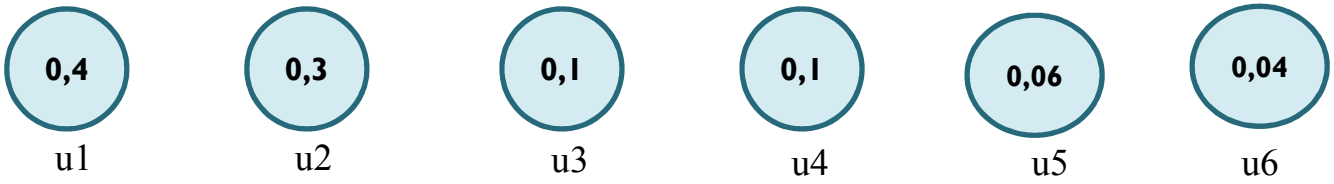
Codage de Huffman

Exemple 1:

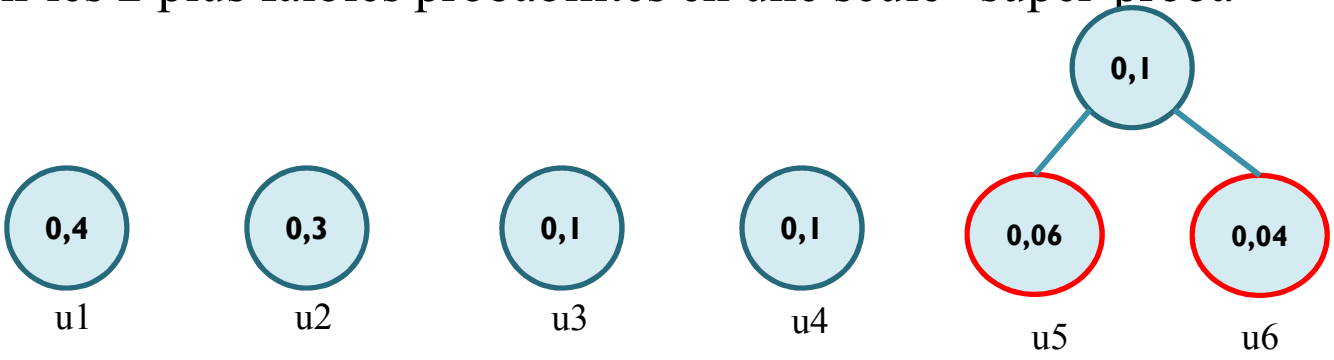
Soit une source émettant les symboles u1,..., u6, avec des probabilités de 0.4, 0.3, 0.1, 0.1, 0.06 et 0.04 respectivement.

Phase 1 : Construction de l'arbre.

Etape 1: Classer les symboles par ordre décroissant de probabilité



Etape2: Réunir les 2 plus faibles probabilités en une seule “super-proba”

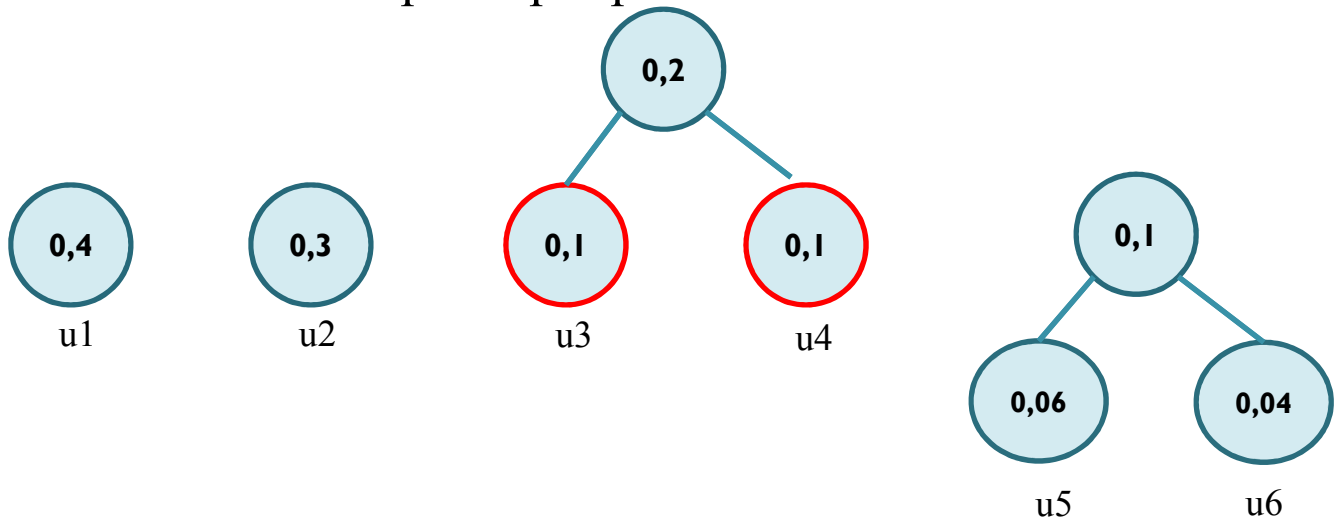


Compression sans perte (codage)

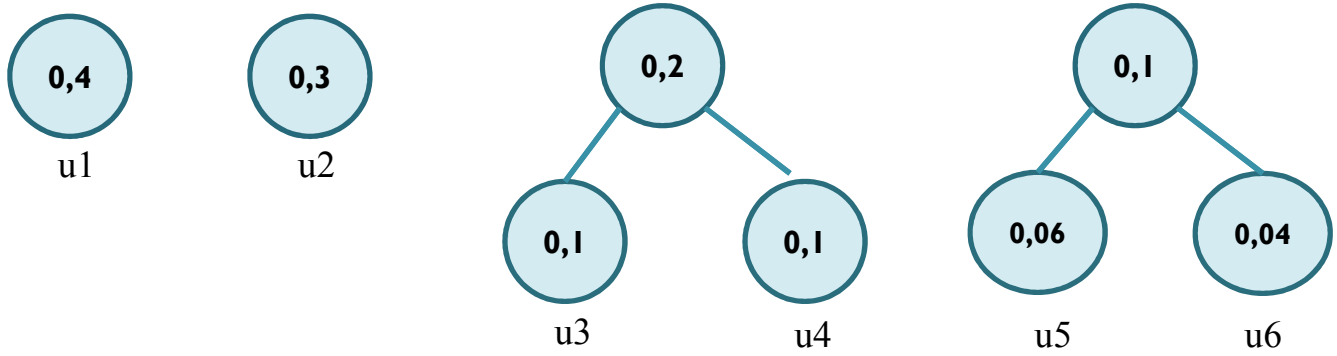
Codage de Huffman

Exemple 1:

Etape 3: Réordonner la table de poids par poids décroissants



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

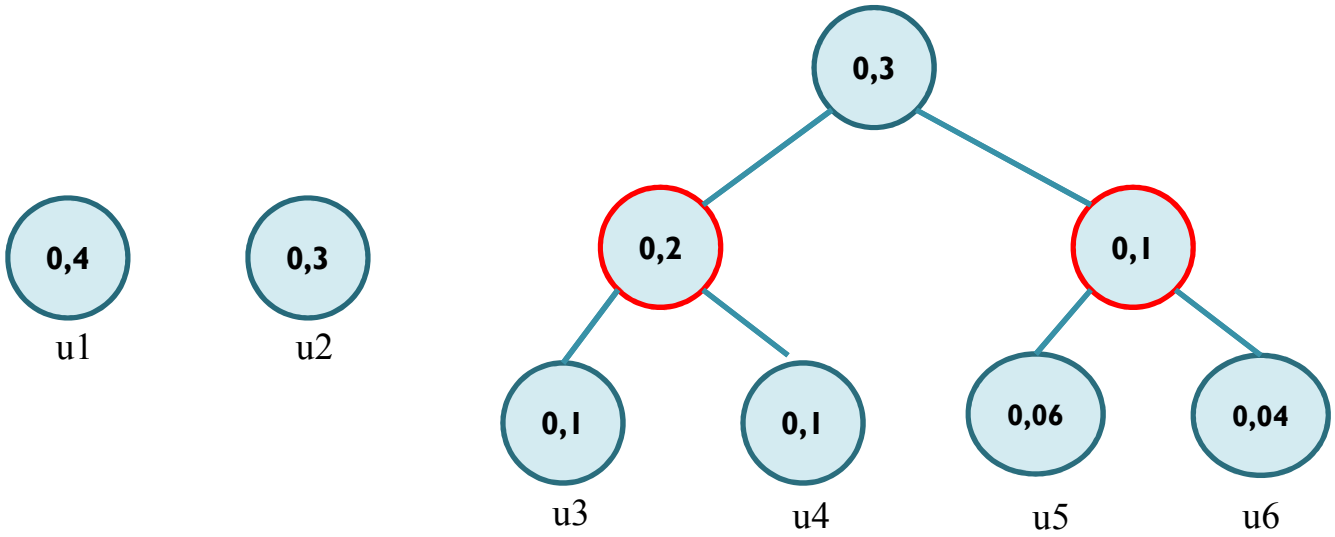


Compression sans perte (codage)

Codage de Huffman

Exemple 1:

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

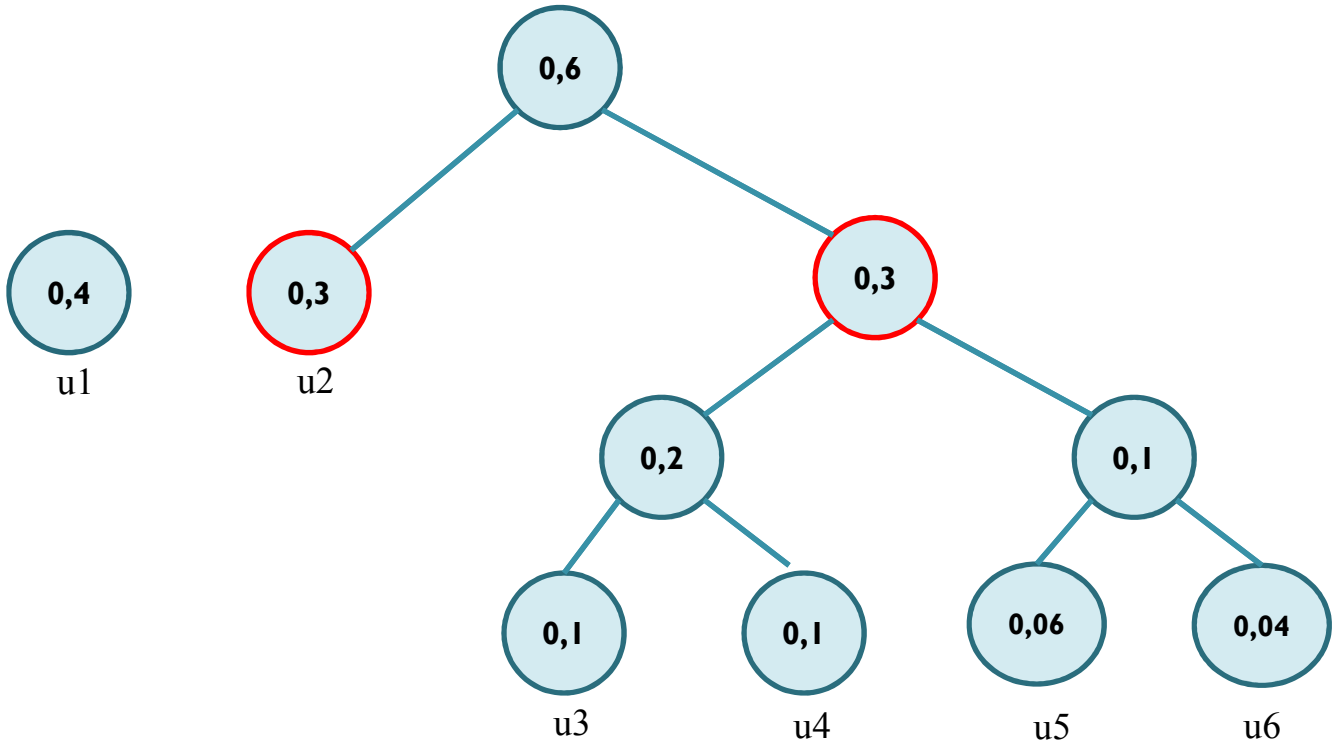


Compression sans perte (codage)

Codage de Huffman

Exemple 1:

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

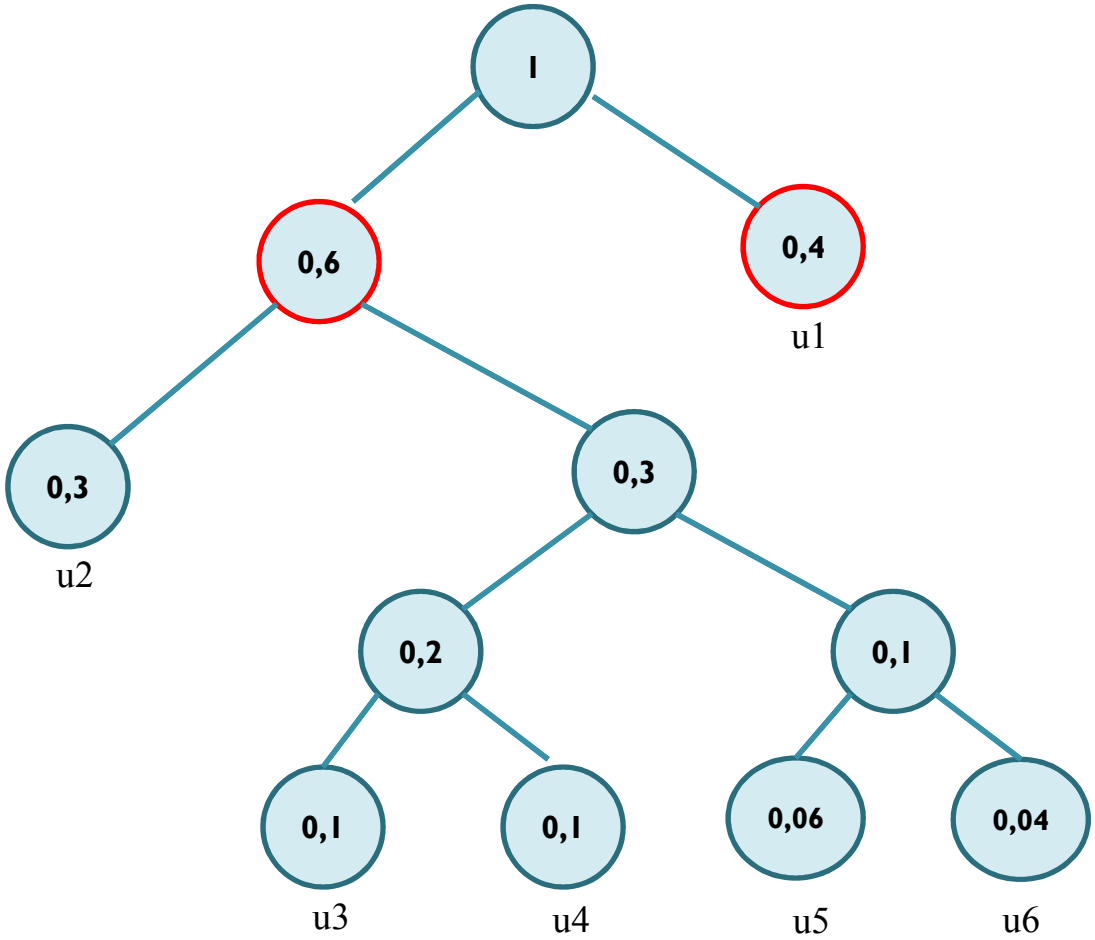


Compression sans perte (codage)

Codage de Huffman

Exemple 1:

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

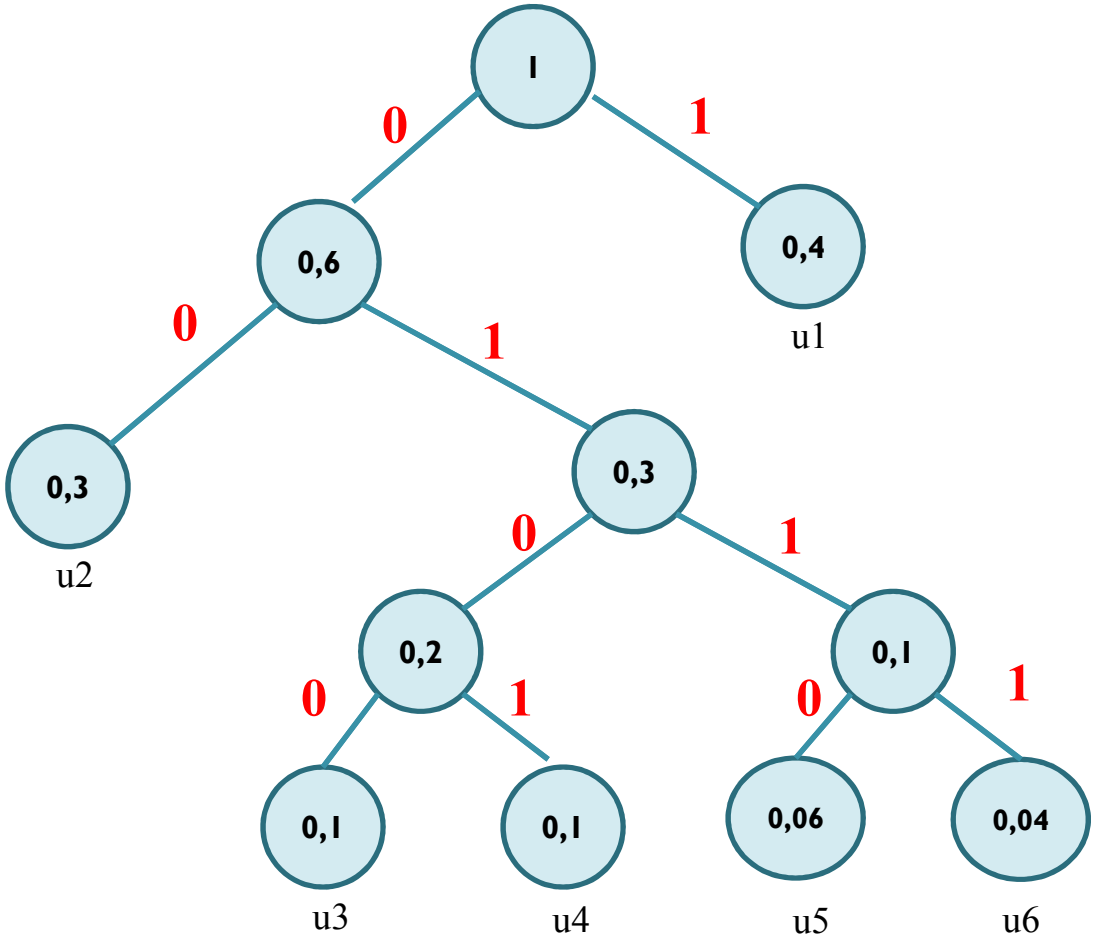


Compression sans perte (codage)

Codage de Huffman

Exemple 1:

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



Compression sans perte (codage)

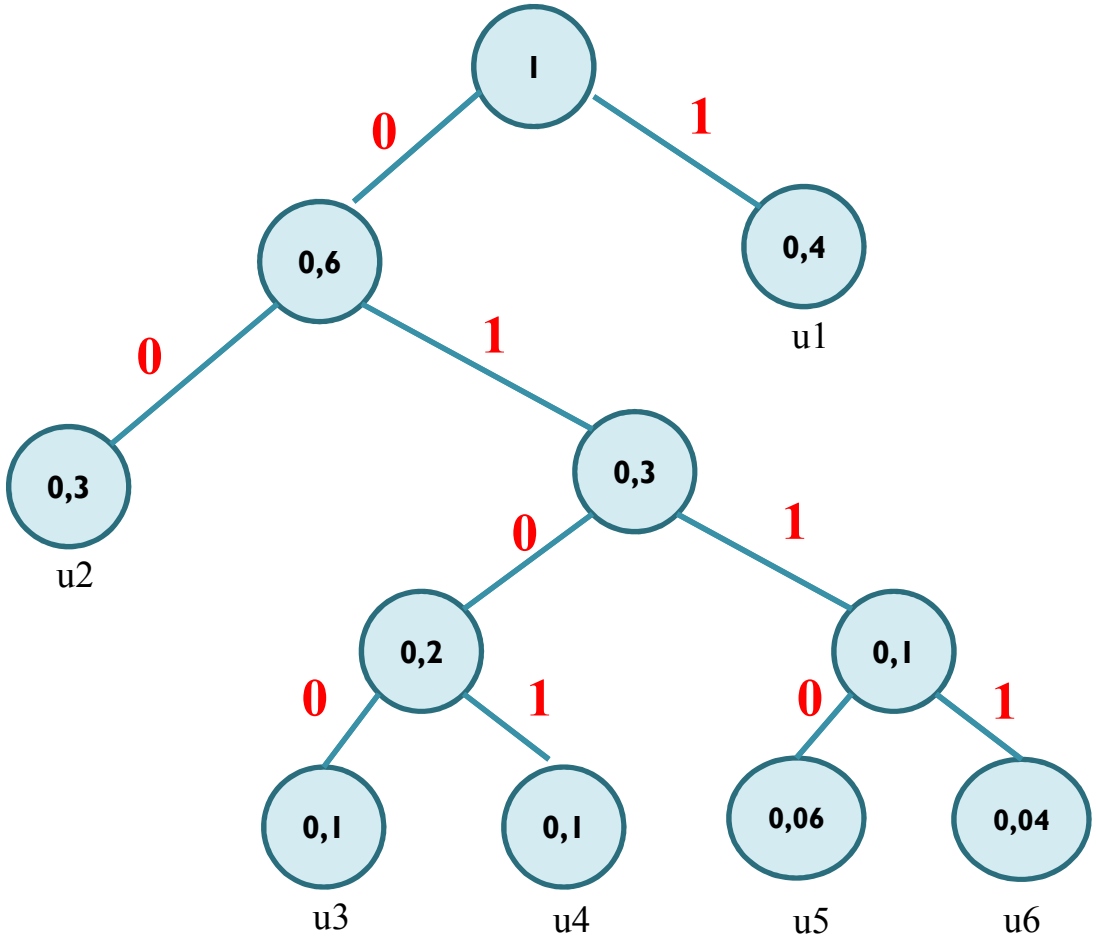
Codage de Huffman

Exemple 1:

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

Lecture du codage obtenu,
du haut vers le bas

Symbole	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’’



Compression sans perte (codage)

Codage de Huffman

Exemple 1:

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

Compression sans perte (codage)

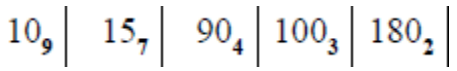
Codage de Huffman

Exemple : compression d'image niveaux de gris

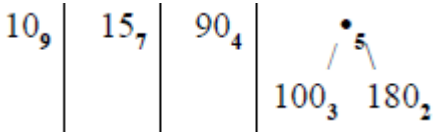
10	15	15	15	15
10	90	100	100	15
10	90	180	100	15
10	90	180	90	15
10	10	10	10	10

➢ **Construction de l'arbre**

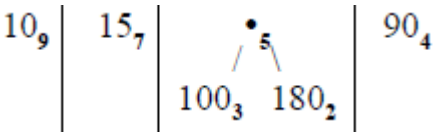
1. Table des poids



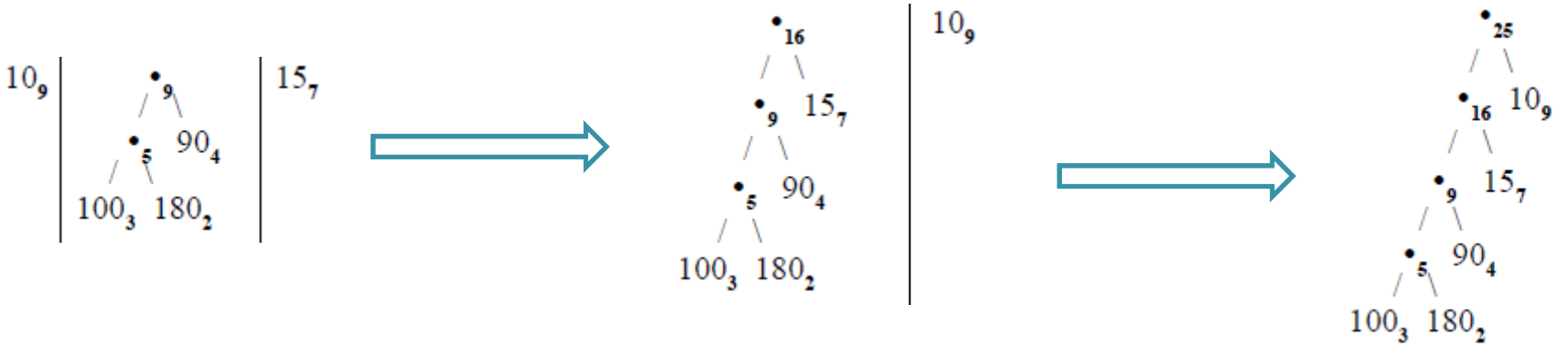
2. Fusion des poids minimaux



3. Réordonnancement



4. Itérations



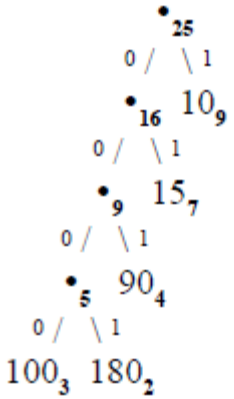
Compression sans perte (codage)

Codage de Huffman

Exemple : compression d'image niveaux de gris

10	15	15	15	15
10	90	100	100	15
10	90	180	100	15
10	90	180	90	15
10	10	10	10	10

➢**Construction du code:** Affectation de valeurs binaires aux arcs



Valeur	Code
10	1
15	01
90	001
100	0000
180	0001

➢**Image codée (en lignes)**

1010101011001000 0 ...

soit 55 bits (image compressée) vs. 25x8=200 bits

Décompression

Le décodeur doit connaître la table de codage (entête) ; extrait les valeurs au plus tôt :

Entrée	Action	Buffer	Émission
1	Identification de 10	vide	10
0	Bufferise et attend	0	rien
1	Identification de 15	vide	15
0	Bufferise et attend	0	rien
1	Identification de 15	vide	15
...	...	...	...

Compression sans perte (codage)

Codage de Huffman

Exercice1

Soit la phrase suivante : *TO BE OR NOT TO BE*

1. Déterminer le nombre d'occurrence de chaque caractère
2. Trier les caractères selon les fréquences d'apparition décroissantes et en cas d'égalité l'ordre alphabétique
3. Appliquer l'algorithme de Huffman
4. Donner le code binaire obtenu,
5. Calculer le quotient, le taux et le gain de compression, conclure.

Exercice 2

Supposons que nous ayons une image en niveaux de gris avec les nombres d'apparitions suivants :

Niveau de gris	128	150	200	220	254
Nbr. app	39	17	16	15	13

1. Construire l'arbre de Huffman correspondant.
2. Donner le code binaire obtenu,
3. Calculer le quotient, le taux et le gain de compression, conclure.

Compression sans perte (codage)

Codage de Huffman

Avantages

- Méthode de compression très **performante**.
- Très bon compromis **temps d'exécution/taux de compression**.
- Algorithme de compression le **plus utilisé**.

Formats utilisant cette méthode

- Images
  - ✓PNG
  - ✓JPEG (Joint Photographic Experts Group) – voir plus loin
- Fichiers compressés
  - ✓Zip
  - ✓Gzip

Compression sans perte (codage)

- RLE (Run-Length Encoding)
- Codage de Huffman
- Méthodes par dictionnaire

Compression sans perte (codage)

Méthodes par dictionnaire: LZW(Lempel-Ziv-Welch)

Définition

LZW(Lempel-Ziv-Welch): Algorithme de compression qui **construit dynamiquement un dictionnaire à partir des données** à compresser, pour **remplacer les chaînes de caractères du dictionnaire par un nouveau code**.

Dictionnaire

Construction

- ✓**Initialisé** avec **les 256 valeurs de la table ASCII**.
- ✓**Découpage** des données en **chaînes d'octets**, comparées **au dictionnaire** et ajoutées si jamais elle n'y est pas présente.

Compression

- ✓**Parcours des données en les codant** ; si jamais une chaîne est plus petite que le plus grand mot du dictionnaire, elle est transmise.

Décompression

- ✓Le dictionnaire est **reconstruit dans le sens inverse**, et n'a donc pas besoin d'être stocké.

Compression sans perte (codage)

Méthodes par dictionnaire: LZW(Lempel-Ziv-Welch)

Avantages

- Méthode **efficace et facile à mettre en place**.
- Plus la **phrase est longue**, plus **les séquences de lettres ajoutées au dictionnaire** sont **grandes**.
- L'algorithme gagne en efficacité avec de grandes données.

Inconvénients

- On **rajoute un bit** pour coder les symboles du dictionnaire tous les **2^n nouveaux symboles**.
- **Si il y a trop de symboles à rajouter** ➔ l'algorithme **n'a plus d'intérêt**.

Formats d'image utilisant cette compression

- TIFF (Adobe).
- GIF .

PLAN

- Introduction
- Compression sans perte (codage)
- Compression avec pertes
- Compression audio/vidéo
- Conclusion

Compression avec pertes

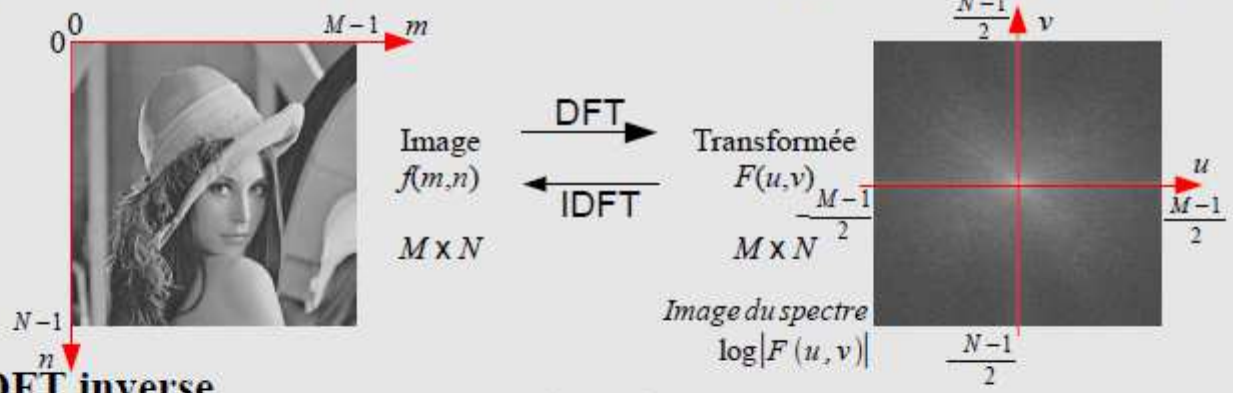
- DCT
- Ondelettes

Compression avec pertes

- DCT (Discret Cosine Transform): Rappel de TFD

Transformée de Fourier discrète 2D (*ang.* DFT) (*rappel*)

- **Notations** Domaine spatial (pixels) Domaine fréquentiel (cycles/pixel)



- **DFT et DFT inverse**

$$F(u,v) := \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-j2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}$$

$$= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \cdot \left[\underbrace{\cos 2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}_{C_{m,n}^{M,N}(u,v)} - j \cdot \underbrace{\sin 2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}_{S_{m,n}^{M,N}(u,v)} \right]$$

$$f(m,n) := \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{+j2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}$$

Compression avec pertes

DCT (Discret Cosine Transform): Rappel de TFD

❖**Inconvénients de la TFD : sur un signal *f réel,***

- Produit un signal **F de spectre symétrique** ; seule la moitié des coefficients
- **Spectraux** a donc besoin d'être **calculée** ;
- Produit un **signal F complexe**, sans que sa partie réelle ou imaginaire seule
- Permette de représenter (donc de reconstruire) le signal f.

❖**DCT est une transformation spectrale (parmi d'autres) qui**

- Possède les **mêmes propriétés que la DFT** ;
- S'applique uniquement sur les **signaux réels** ;
- Est définie par des fonctions de base en ***cosinus seulement*** ;
- Est utilisée en **compression d'images /vidéo(JPEG, MPEG).**

Compression avec pertes

DCT (Discret Cosine Transform):

Définition

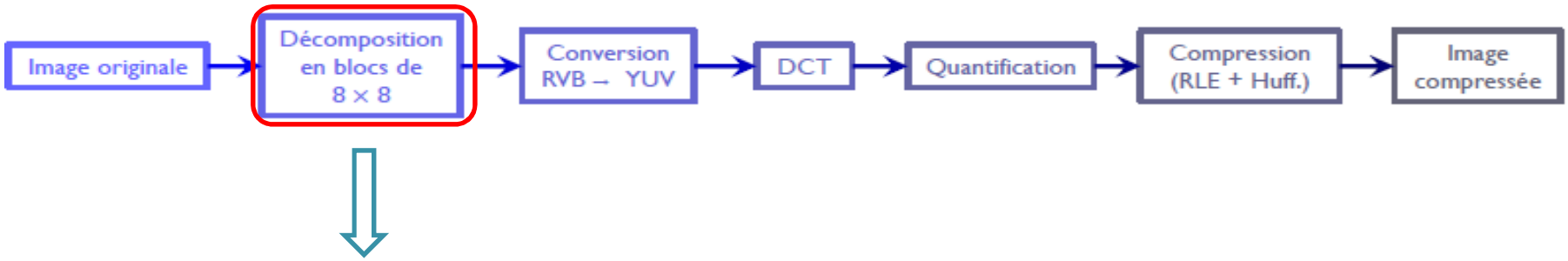
- Algorithme de compression basé sur la transformée DCT (*Discret Cosine Transform, en français transformée en cosinus discrète),variante de la transformée de Fourier.*
- Cette méthode permet de décrire chaque **bloc** en une carte de **fréquences** et en **amplitudes** plutôt qu'en **pixels** et **couleurs**.

Remarques

- Fréquence→ **importance et rapidité** d'un **changement de couleur**
- Amplitude→ **écart** pour chaque **changement de couleur**
- Utilisée pour **les formats JPEG et MPEG**.

Compression avec pertes

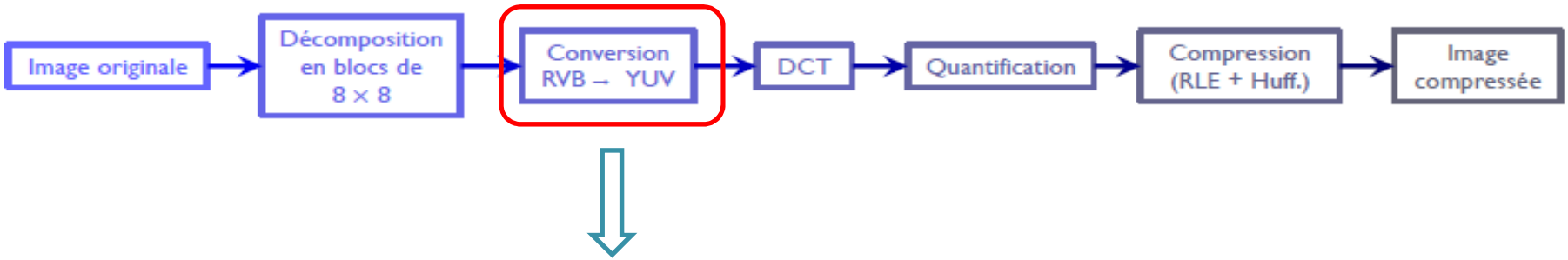
DCT (Discret Cosine Transform): étapes de compression



- **L'image** est divisée en **blocs de 8×8 pixels**
- Prolongement éventuel de l'image
  - Par des zéros.
  - Par symétrie.
  - …
- Chaque **bloc est une 'petite' image**

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression

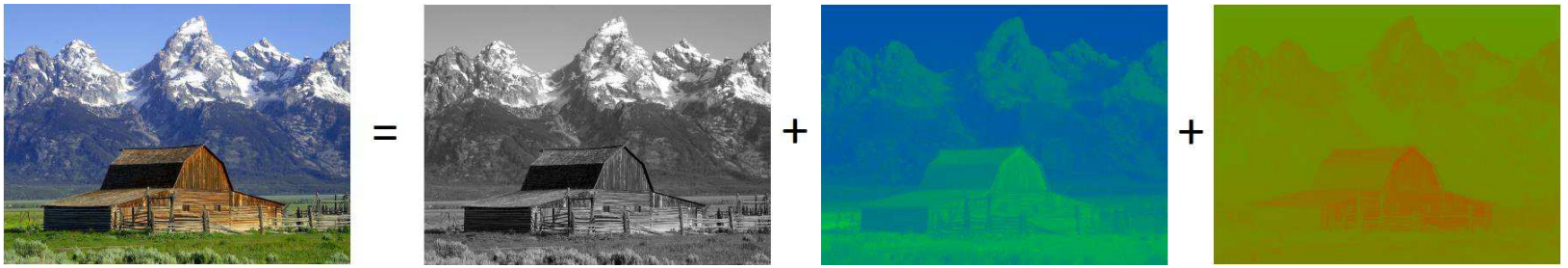


- Conversion des couleurs dans l'espace YUV
  - ✓Y est une composante de **luminance**.
  - ✓U et V sont deux **composantes de couleurs**.
  - ✓Conversion réversible : RVB ↔ YUV

$$Y = 0{,}299\,\mathbf{R} + 0{,}587\,\mathbf{V} + 0{,}114\,\mathbf{B}$$

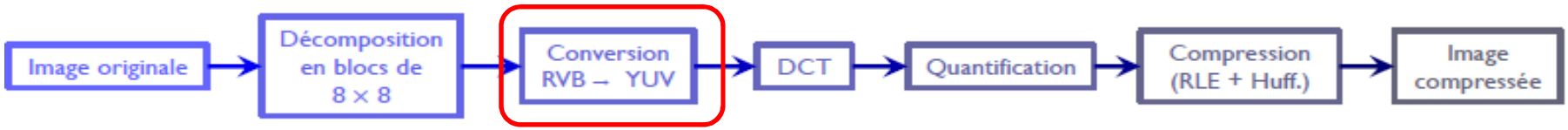
$$U = -0{,}1687\,\mathbf{R} - 0{,}3313\,\mathbf{V} + 0{,}5\,\mathbf{B}$$

$$V = 0{,}5\,\mathbf{R} - 0{,}4187\,\mathbf{V} - 0{,}0813\,\mathbf{B}$$



Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression



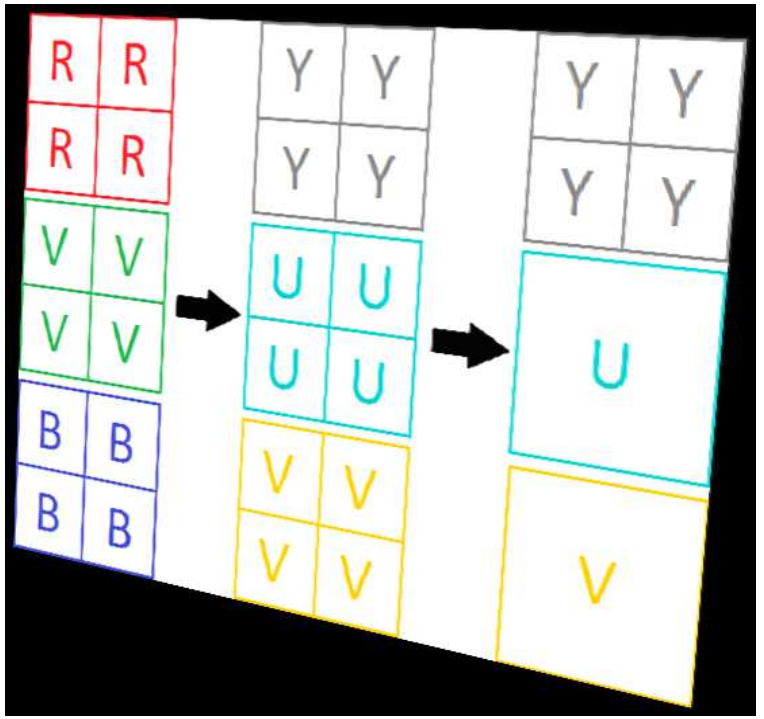
On exploite le fait que l'oeil humain est plus sensible à l'intensité lumineuse qu'aux informations colorées.

✓On réduit les **composantes de chrominances par 4**.

✓On réduit l'information de **chaque bloc par 2**.

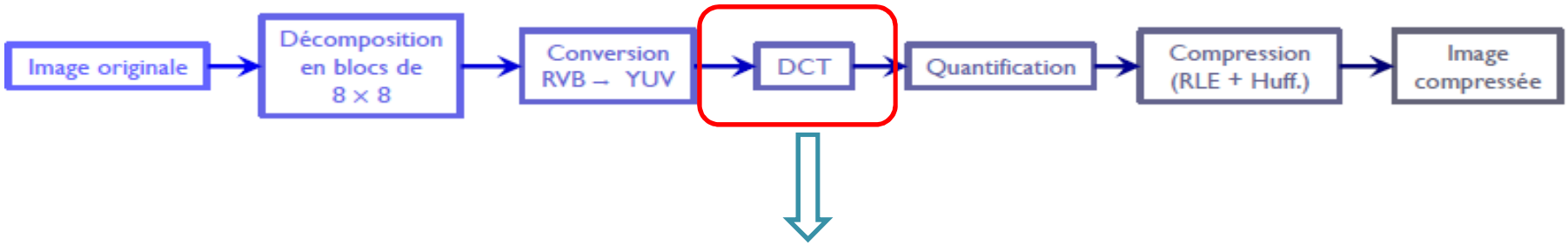
✓**Moins de valeurs** : **image plus légère**.

✓**Perte d'informations : dégradation de l'image**.



Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression



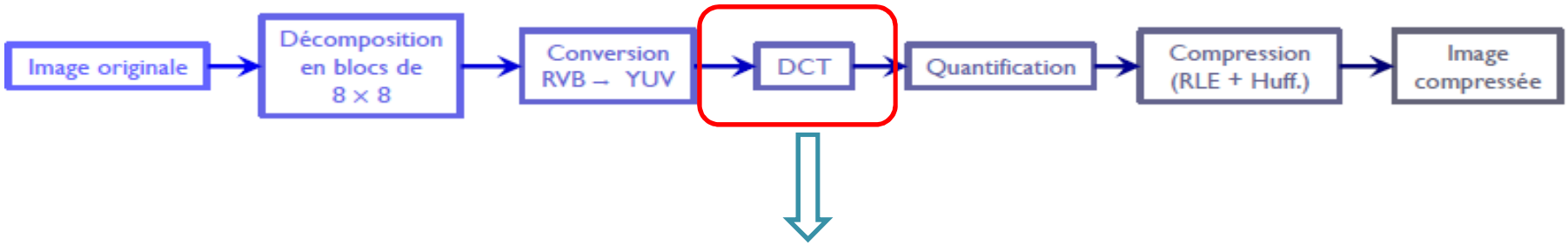
- On applique la DCT sur chaque bloc (matrice) de 8×8
  - ✓ Chaque **matrice** est transformée en une **autre matrice**.
  - ✓ Il n'y a pas de perte d'informations.
  - ✓ La **transformée inverse existe**.

$$DCT(i,j) = \frac{1}{\sqrt{2}} C(i)C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} pixel(x,y).cos\frac{(2x+1)i\pi}{16}.cos\frac{(2y+1)i\pi}{16}$$

$$\begin{cases} C(x) = \frac{1}{\sqrt{2}} \;\; si\; x = 0 \\ C(x) = 1 \;\; si\; x > 0 \end{cases}$$

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression



- Chaque **sous-bloc en 8x8** est transformée en **une autre matrice par la DCT**

$$I_0 = \begin{matrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 160 & 161 & 162 & 162 & 155 & 155 & 155 & 161 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{matrix}$$

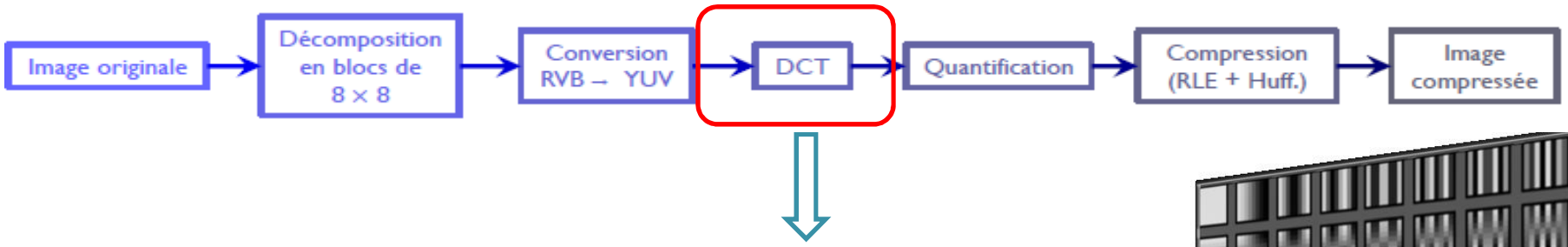
Matrice originale

$$I_1 = \begin{matrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{matrix}$$

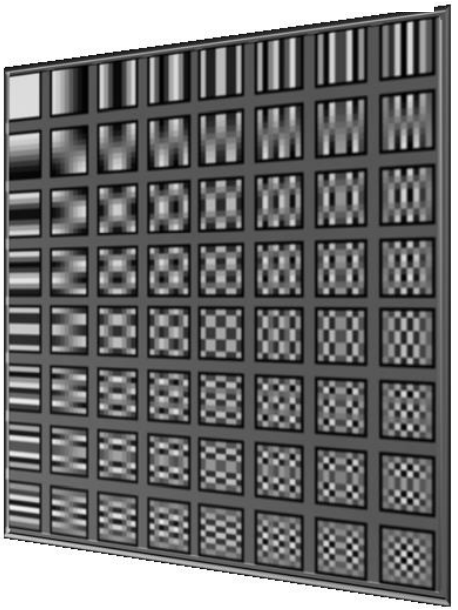
Matrice transformée par DCT

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression

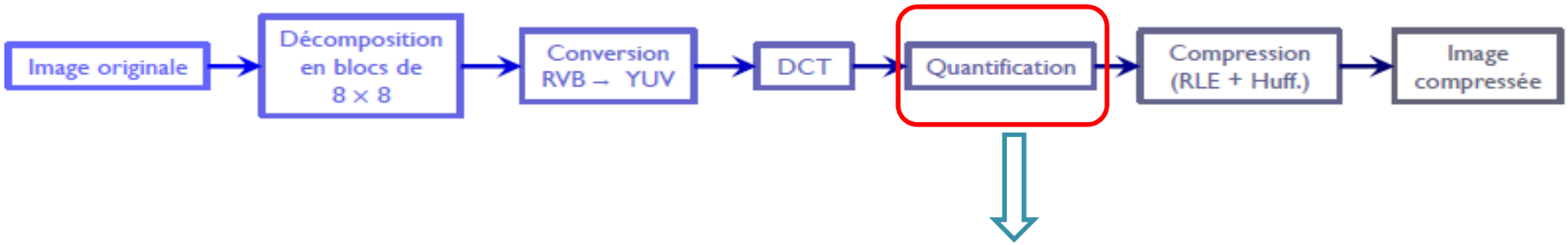


- C'est une représentation **fréquentielle**
  - ✓ Composante continue ***au coin en haut à gauche***
  - ✓ Basses fréquences ***en haut à gauche***
  - ✓ Hautes fréquences ***en bas à droite***



Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression



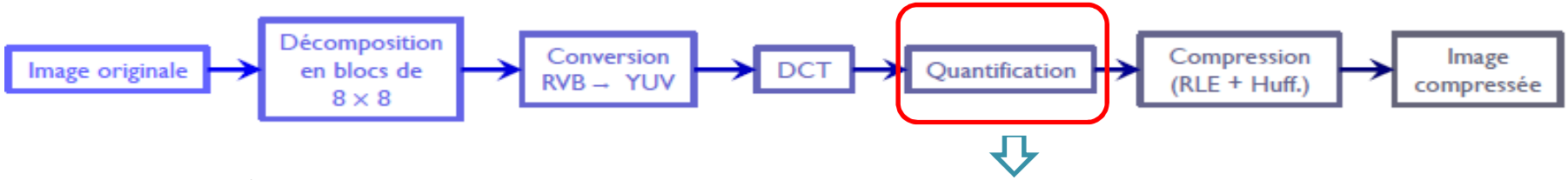
But

➢**Atténuer les hautes fréquences** (auxquelles l'oeil humain est peu sensible) en **mettant les coefficients à 0**.

- Processus principalement **responsable de la dégradation de l'image**.
- Elle permet **une compression (étape suivante) très efficace**.
- C'est ici que la **majorité de l'information est perdue**.
- Elle **est réversible mais induit des pertes**.

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression



➢Grâce à un **facteur de qualité**, on crée une table de quantification 8x8 selon la formule : $Q(i,j) = 1 + (i + j + 1) \times q$

➢Exemple pour un facteur de qualité de (**q=3**) :

$$I_1 = \begin{matrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{matrix}$$

Matrice transformée par DCT

$$Q = \begin{matrix} 4 & 7 & 10 & 13 & 16 & 19 & 22 & 25 \\ 7 & 10 & 13 & 16 & 19 & 22 & 25 & 28 \\ 10 & 13 & 16 & 19 & 22 & 25 & 28 & 31 \\ 13 & 16 & 19 & 22 & 25 & 28 & 31 & 34 \\ \vdots & & & & & & & \\ 22 & 25 & 28 & 31 & 34 & 37 & 40 & 43 \\ 25 & 28 & 31 & 34 & 37 & 40 & 43 & 46 \end{matrix}$$

$Q(2,4) = 1 + (2 + 4 + 1) \times 3$

Matrice de quantification de qualité 3

$$I_2 = \begin{matrix} 315 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Matrice quantifiée

$$I_2(i,j) = \left\lfloor \frac{I_1(i,j)}{Q(i,j)} \right\rfloor$$

= La valeur d'un élément de la matrice DCT quantifiée sera **égale à l'arrondi, à l'entier le plus proche**, du quotient de la valeur correspondante de la matrice DCT par la valeur correspondante de la matrice de quantification

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression

$I_2 =$

315	0	−1	0	0	0	0	0
0	−2	0	0	0	0	0	0
−1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Matrice quantifiée

1	315	2	0	1	−1	1	−2	1	−1	58	0

Matrice compressée (RLE)

- Chaque matrice de 8×8 est compressée
  - ✓ Une première compression **RLE**.
  - ✓ Deuxième avec le codage **d'Huffman**.
- Etape réversible.
- Particularité
  - ✓ Les coefficients sont lus en **zig-zag**.
  - ✓ La compression RLE est optimisée.

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression

En bref

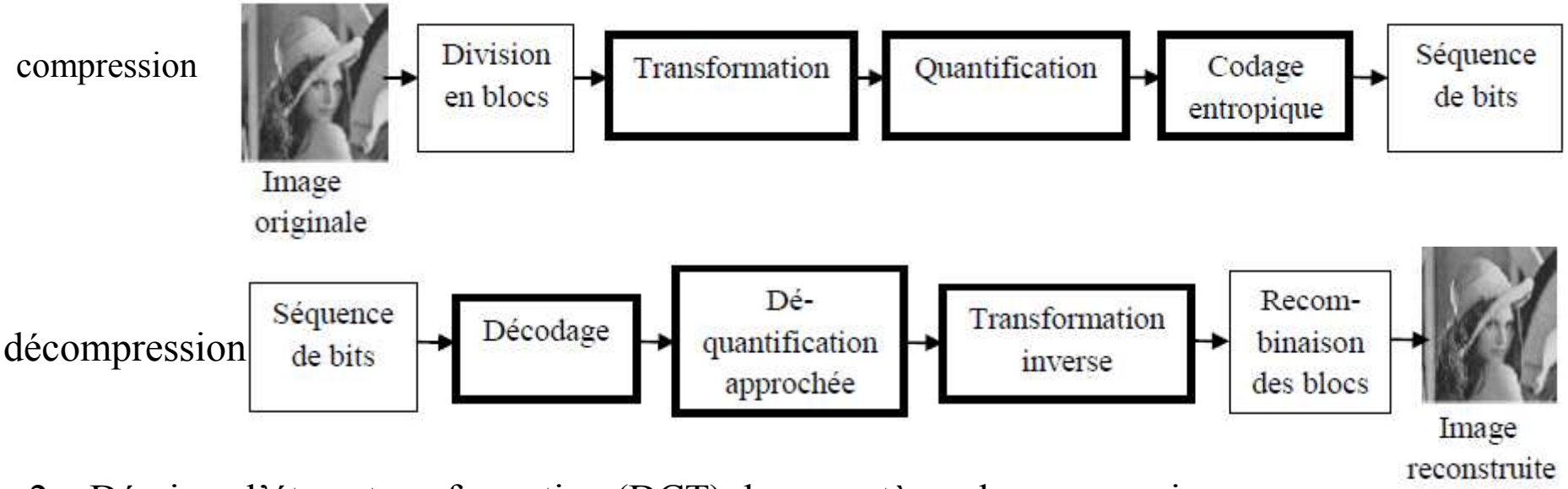
- Compression réalisée en **supprimant les hautes fréquences** (détails et contours).
  - ✓**Perte** d'information peu visible.
  - ✓**Bonne** compression.

- Formats d'image utilisant cette compression
  - ✓**JPEG**.

Compression avec pertes

DCT (Discret Cosine Transform): Exercice 1

1. Donnez le schéma d'un système de compression/décompression **JPEG .**



2. Décrivez l'étape transformation (DCT) de ce système de compression

Elle consiste à appliquer une transformation mathématique à chaque bloc. Les transformations utilisées en compression d'image sont des transformations orthogonales, leur but est de décorréler les pixels, ce qui a pour effet en général de redistribuer l'énergie de l'image dans un nombre restreint des coefficients transformés.

Compression avec pertes

DCT (Discret Cosine Transform): Exercice 1

3. Quel effet apporte la DCT sur les coefficients de la matrice d'entrée

La DCT a pour effet de concentrer l'information en très peu de coefficients fréquentiels correspondant aux basses fréquences, et que les autres coefficients sont de haute fréquence. Dans la matrice, suite à l'application de la DCT, les basses fréquences se trouvent en haut à gauche et les hautes fréquences en bas à droite. Les hautes fréquences représentent les zones à forts contrastes dans l'image,

4. Quel l'intérêt de la quantification dans ce système de compression

La quantification permet de mettre à zéro tous les coefficients inférieurs au quantum de la table, et donc de mettre à zéro une grande partie des hautes fréquences. Car la grande partie de l'information de l'image est stockée dans les basses fréquences après l'application du TCD

Compression avec pertes

DCT (Discret Cosine Transform): Exercice 2

On cherche à coder l'image suivante en JPEG.

100	150	150	100
100	100	150	150
200	100	100	150
100	200	100	100

La matrice DCT obtenue est la matrice 4x4 ci-dessous :

512	7	-13	-16
-7	-39	-20	44
-13	20	-87	-49
16	44	49	14

On utilise la matrice de quantification définie par $Q=[q_{i,j}]$ avec $q_{i,j}=1+\kappa(1+i+j), i\in[\![0,N-1]\!], j\in[\![0,N-1]\!]$,
N la taille du bloc

1. Ecrire la matrice de quantification pour les facteurs de qualité k=9 et k=20.
2. Calculer les valeurs de la matrice DCT quantifiée pour les deux cas.
3. Donner la suite générée après la lecture zigzag pour les deux cas.
4. Donnez le codage RLE pour cette image dans les deux cas. Conclure.

Compression avec pertes

DCT (Discret Cosine Transform): Exercice 3

On cherche à coder une image (4x4) en JPEG. On a obtenu la matrice DCT avec les valeurs indiquées dans la matrice 4x4 ci-dessous.

1. Donner la matrice de quantification classique de facteur $\kappa = 4$.
2. Donner la matrice-résultat après quantification.
3. Donner la suite générée après la lecture zigzag.
4. Donner son codage en Run Length Encoding.

1120	82	-48	36
56	8	32	4
-208	-40	22	-12
30	8	-8	2

Compression avec pertes

DCT (Discret Cosine Transform): Exercice 4

On cherche à coder l'image suivante en JPEG.

Matrice source (Non signée) :

29	95	60	60	64	52	61	63
123	85	119	114	118	103	95	103
65	79	82	106	115	95	97	73
71	79	99	82	119	122	96	106
106	86	90	148	132	132	116	162
93	99	79	103	123	113	115	117
141	155	88	102	108	95	118	103
134	160	93	104	97	102	105	128

La matrice DCT obtenue est la matrice 4x4 ci-dessous :

Matrice DCT :

-214	-23	-4	8	18	-27	-24	-12
-103	-15	-52	-16	-8	8	19	19
-40	66	41	-11	-24	-13	-30	-6
-20	-19	4	-7	8	-3	-9	-7
-21	-19	9	-20	-4	-20	7	-17
-68	11	0	6	-37	-18	-23	8
-58	-19	-17	-4	-13	-10	-18	-5
4	1	-2	-6	13	8	-21	-17

On utilise la matrice de quantification définie par $Q=[q_{i,j}]$ avec $q_{i,j}=1+\kappa(1+i+j), i\in[\![0,N-1]\!], j\in[\![0,N-1]\!]$,

N la taille du bloc

1. Ecrire la matrice de quantification pour les facteurs de qualité k=2 et k=20.
2. Calculer les valeurs de la matrice DCT quantifiée pour les deux cas.
3. Donner la suite générée après la lecture zigzag pour les deux cas.
4. Donnez le codage RLE pour cette image dans les deux cas. Conclure.
5. Donnez le codage de Huffman.

Compression avec pertes

- DCT (Discret Cosine Transform): Exercice 3 (solution)

Matrice de quantification :

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Matrice DCT quantifiée :

-71	-4	0	0	1	-2	-1	0
-20	-2	-5	-1	0	0	1	1
-5	7	3	0	-1	0	-1	0
-2	-1	0	0	0	0	0	0
-1	-1	0	-1	0	0	0	0
-5	0	0	0	-1	0	0	0
-3	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Développement en zig-zag :

-71 -4 -20 -5 -2 0 0 -5 7 -2 -1 -1 3 -1 1 -2 0 0 0 -1 -5 -3 0 0 0 -1 0 -1
0 -1 0 -1 0 0 0 0 0 0 -1 1 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

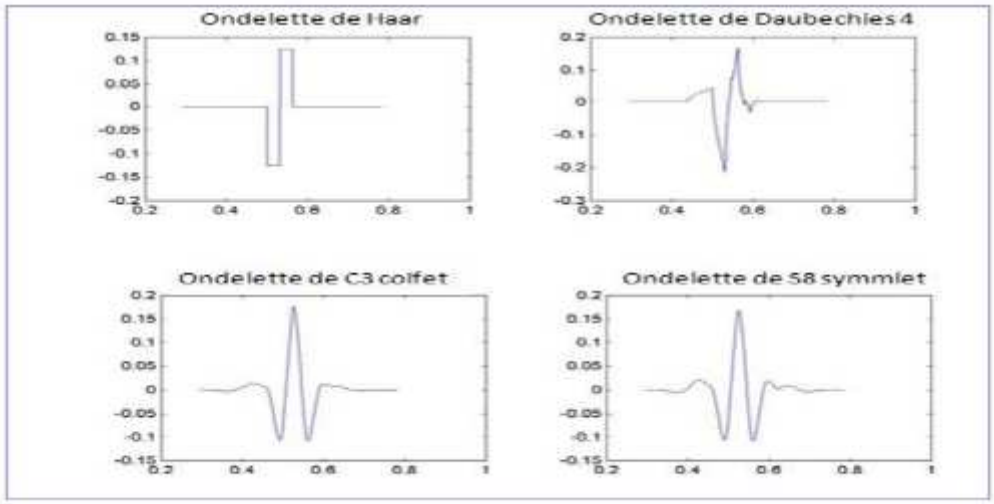
Compression avec pertes

- DCT
- Ondelettes

Compression avec pertes

Transformée en ondelettes: Définition

- Une **ondelette** est une **forme d'onde** qui a une **valeur moyenne zéro** et une **durée limitée.**
- Transformée en ondelettes (**Wavelet Transform**); est un **outil mathématique** qui **décompose** un signal en **fréquences** en conservant une localisation spatiale. **Le signal de départ est projeté** sur un **ensemble de fonctions de base** qui varient en fréquence et en espace.



Quelques familles d'ondelettes

- **Séparer** de **l'information général** (basses fréquences) et les **détails** (hautes fréquences), contenue dans une image, un son, …
- Utilisée pour les formats **JPEG2000 et MPEG2000**.

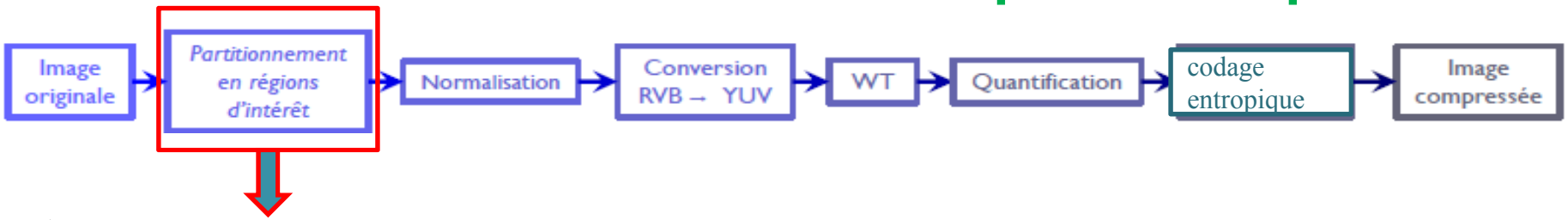
Compression avec pertes

Transformée en ondelettes: domaines d'application

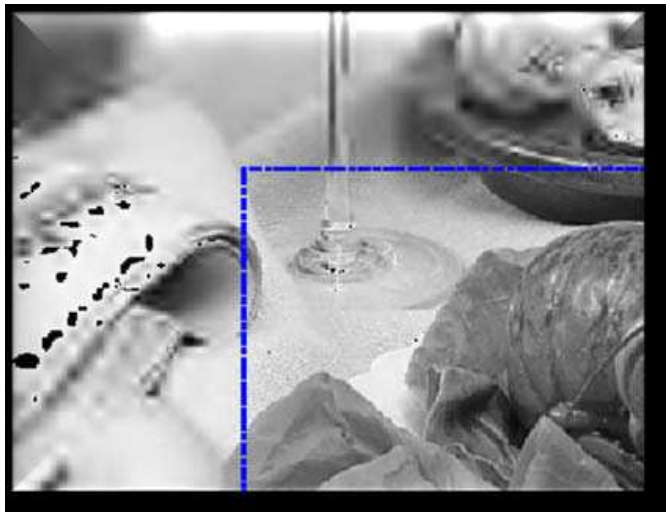
1. **Compression des données de haute résolution (e.g.** images, vidéo)
2. **Filtrage (filtering, de-noising)**
3. **Lissage (smoothing)**
4. **Extraction des propriétés caractéristiques (Feature Extraction)**
5. **Détection des discontinuités (discontinuity detection)**
6. **Analyse des données (data analysis)** (par exemple, biomédicales, financières)
7. **Télécommunication (par exemple, codage de sources et canaux (Source and Channel Coding))**
8. **Astronomie** (par exemple, distances dans l'univers, galaxies formant de structures hiérarchiques à différent niveaux de l'échelle).
9. **Analyse de séries temporelles pour des prévisions de marché boursier**.
10. **Réseaux d'ondelette (wavelet networks) : apprentissage en temps réel des fonctions inconnues.**
11. **….**

Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



- Etape **facultative**
- On y définit des **zones (régions d'intérêt ou ROI)**
  - ✓ La qualité de ces zones est préservée.
  - ✓ Ces régions sont **mieux encodées**, au détriment du reste de l'image.

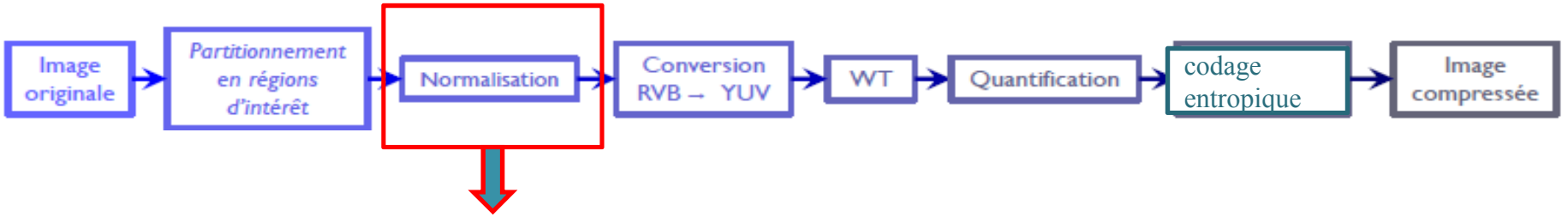


Avec ROI

Sans ROI

Compression avec pertes

- Transformée en ondelettes discrètes : **étapes de compression**

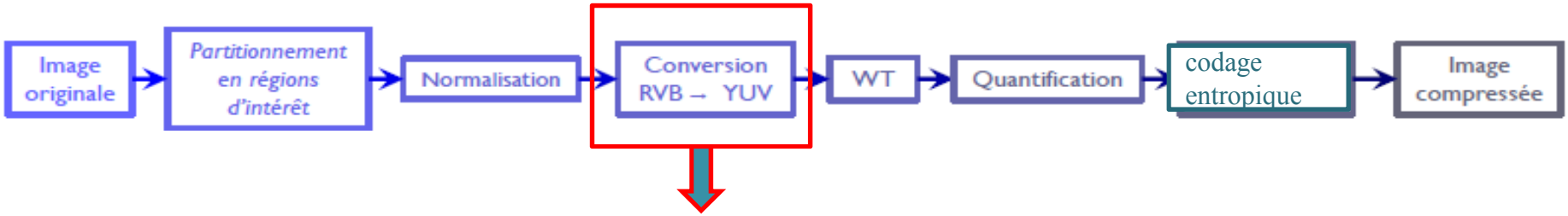


Décalage des valeurs de l'intervalle **[0;255] vers [−128;127]**

- But **: centrer les valeurs autour de 0**.
- Etape **nécessaire** à une **bonne compression**

Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression

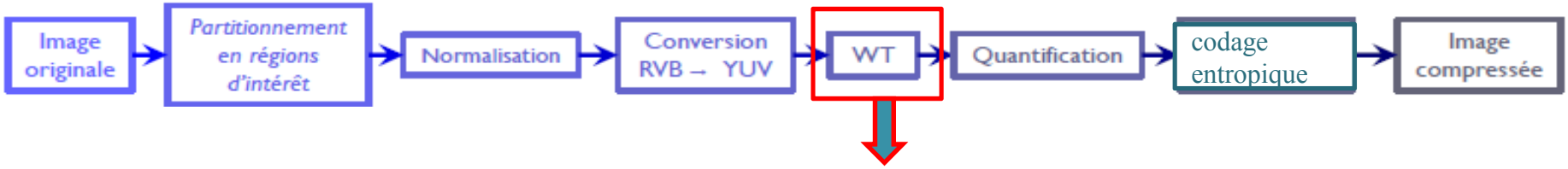


- Même **conversion que pour la compression utilisant la transformée en cosinus discrètes.**

- Synthèse de couleurs mieux adaptée à la compression car les composantes sont moins corrélées.

Compression avec pertes

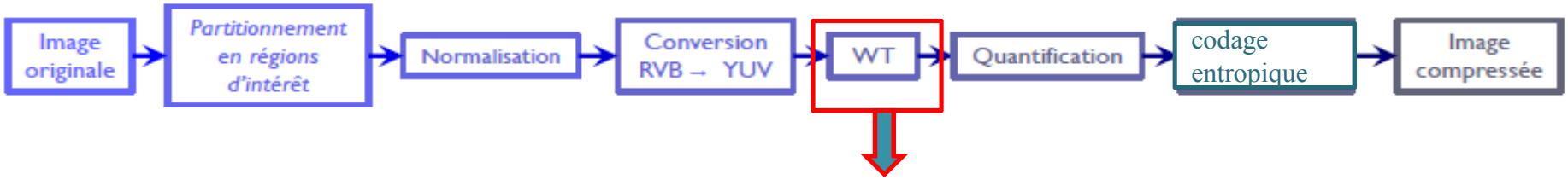
Transformée en ondelettes discrètes : étapes de compression



- N'importe quelle **transformée en ondelettes peut être utilisée**
- Les plus utilisées
  - ✓ Ondelette de **Daubechies**.
  - ✓ Ondelette de Le **Gall**.
- Exemple des ondelettes les plus simples : **ondelettes de Haar**
  - ✓ Permet de **décomposer une séquence en basses et hautes fréquences**

Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



→Exemple: Interprétation (transformée en ondelettes 1D de Haar)

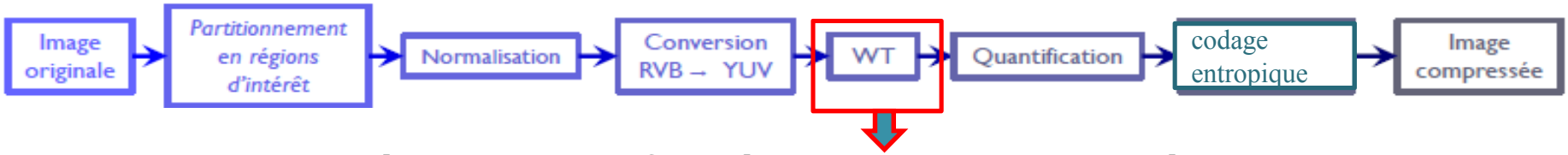
1. On calcule d'abord **les moyennes 2 à 2**, Ces termes représentent de manière grossière (**moyenne**) de la **séquence**
2. On calcule ensuite les **différences 2 à 2**, Ces termes de différences sont aussi appelés les **détails de la séquence.**
3. On **réitère** ces **2 opérations** sur les **coefficients de moyenne de la nouvelle séquence** jusqu' on trouve une **seule valeur moyenne**.

Niveau de détail #	Coefficients de moyenne	Coefficients de détail
0	[9 ; 7 ; 3 ; 5]	
1	L1 = [8 = (9+7)/2 ; 4 = (3+5)/2]	H1 = [1 = (9-7)/2 ; -1 = (3-5)/2]
2	L2 = [6 = (8+4)/2]	H2 = [2 = (8-4)/2]

- L : **filtre passe-bas** (***Low pass filter***), **H : filtre passe-haut** (***High pass filter***)
- Séquence originale **: [9 ; 7 ; 3 ; 5],** Séquence finale **: [6 ; 2 ; 1 ; -1]**

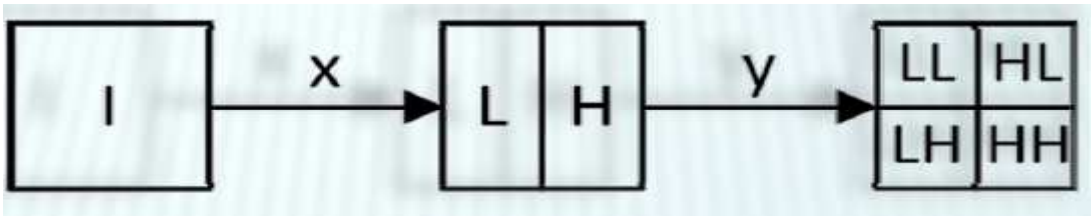
Compression avec pertes

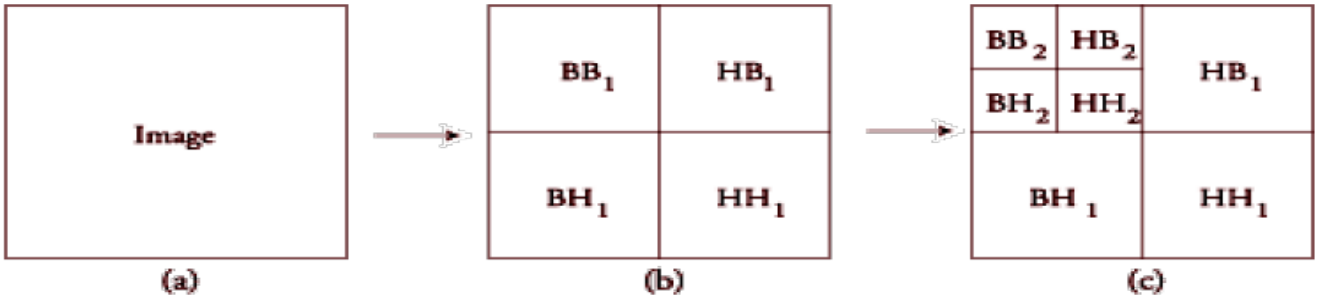
Transformée en ondelettes discrètes : étapes de compression



→Exemple: Interprétation (Transformée en ondelettes discrètes 2D de Haar)

➢ **Construction par succession d'ondelettes 1D discrètes suivant les axes x, puis y de l'image 2D**



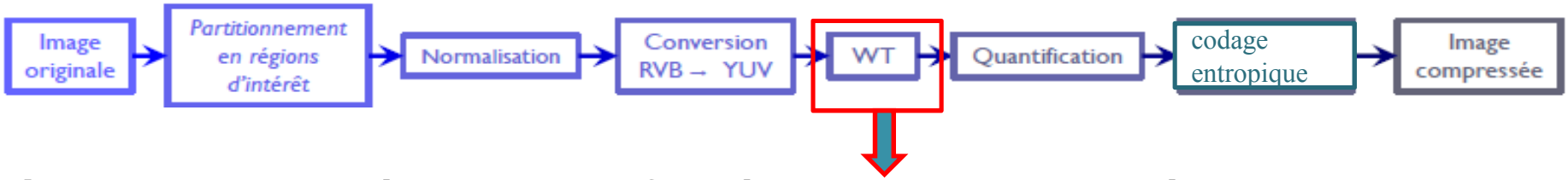


4 niveaux de détail de la décomposition en ondelettes 2D

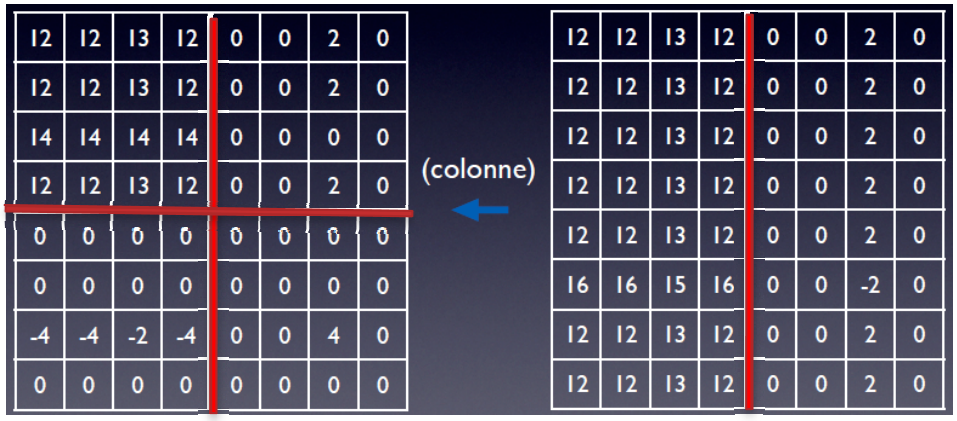
Pour chaque niveau de détail, ***l'information générale est en haut à gauche****, et* ***les détails dans la zone restante***

Compression avec pertes

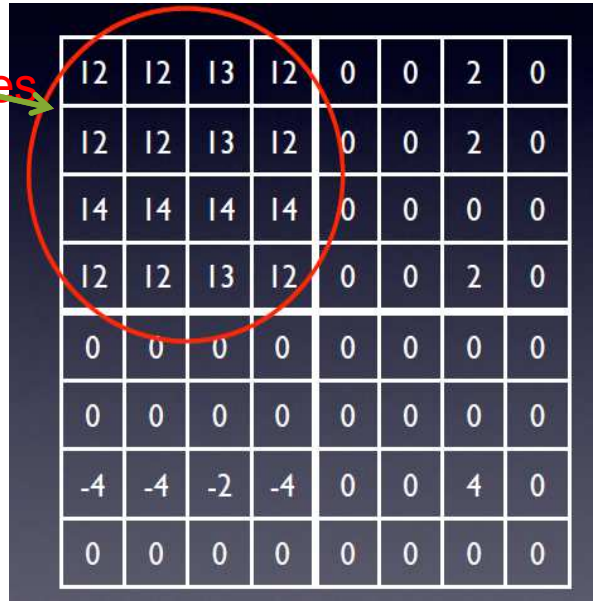
Transformée en ondelettes discrètes : étapes de compression



→Exemple: Interprétation (Transformée en ondelettes discrètes 2D de Haar)



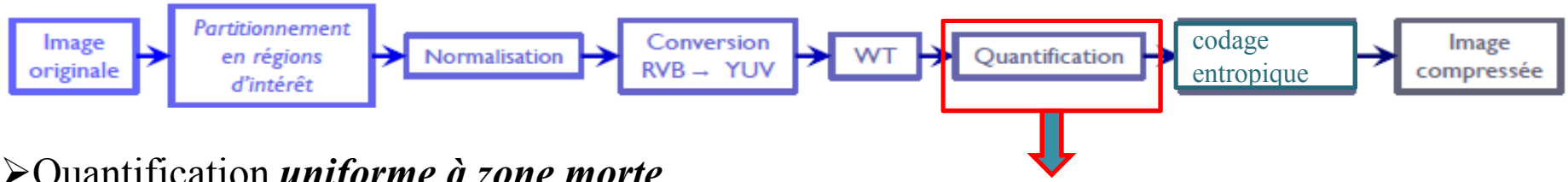
Basses fréquences



Reste = hautes fréquences, qui pourront être fortement quantifiées

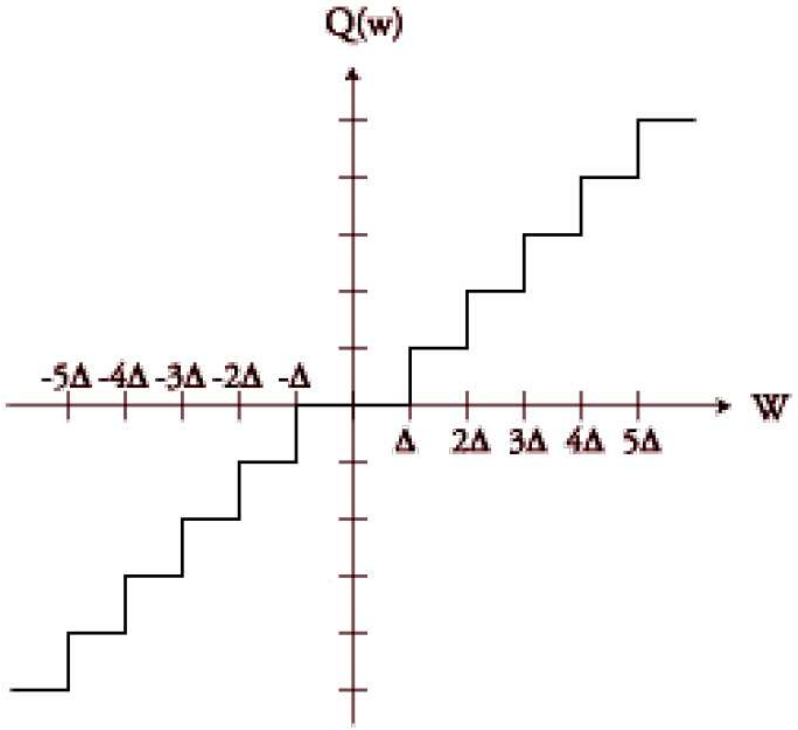
Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



- Quantification ***uniforme à zone morte***.
  - ✓La **courbe** évolue de manière **uniforme**.
  - ✓Les valeurs quantifiées peuvent être égales à zéro.

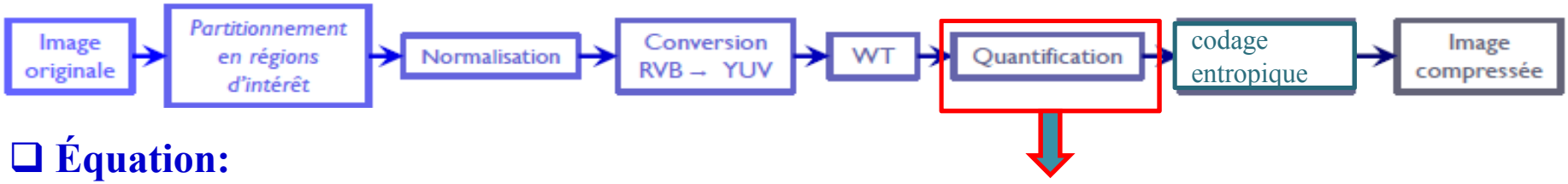
- Deux caractéristiques
  - ✓Le **pas Δ** : l'intervalle pour lequel on a une même valeur quantifiée.
  - ✓La **largeur** de la zone morte ***Z*** : **l'intervalle** pour lequel la **valeur quantifiée** est **nulle**.



Attention : la quantification ne s'applique qu'aux coefficients de détail, et non pas au coefficient de moyenne.

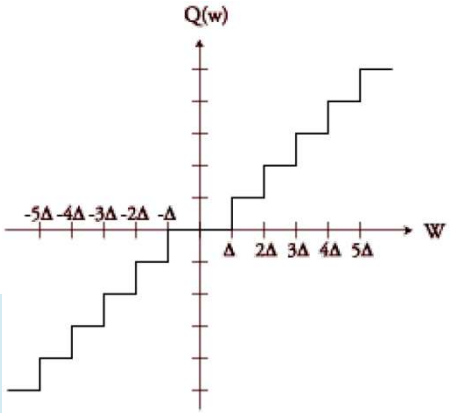
Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



❑ Équation:

- Étant donné que l'on a une zone morte de largeur **Z**, cela signifie **que si un coefficient de détail C, en valeur absolue**, est dans la zone morte, c'est-à-dire **| C | < Z**, alors le coefficient est **mis à 0**.
- Sinon, si un coefficient de détail C, en valeur absolue, est hors de la zone morte, c'est-à-dire | C | ≥ Z, il sera quantifié selon le pas de quantification **Δ** choisi.

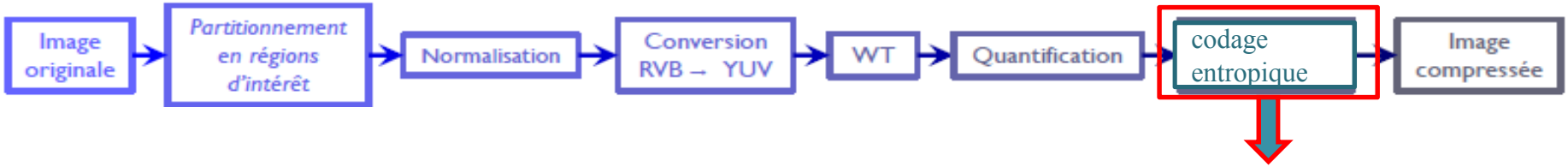


❑ L'équation de quantification finale est donc :

- **Si | C | < Z, alors $C_{résultant} = 0$**
- **Si | C | ≥Z, alors $C_{résultant}$ = signe(C) × Δ × partie_entière (| C | / Δ)**

Compression avec pertes

- Transformée en ondelettes discrètes : **étapes de compression**



➢ Coder le résultat obtenu à l'aide de Huffman

Compression avec pertes

Transformée en ondelettes discrètes : **Exercice 1**

1. Soit une séquence d'origine S_0 = [2 4 8 12 14 0 2 1], correspondant aux données initiales. Donner la décomposition en ondelettes 1D de Haar de la séquence S_0 .

le tableau de la décomposition en ondelettes 1D de Haar de S0 est le suivant:

Niveau de résolution n	Coefficients de moyenne S_n	Coefficients de détail D_n
0 – séquence d'origine	S_0 = [2 4 8 12 14 0 2 1]	----------
1	S_1 = [3 10 7 1.5]	D_1 = [–1 –2 7 0.5]
2	S_2 = [6.5 4.25]	D_2 = [-3.5 2.75]
3	S_3 = [5.375]	D_3 = [1.125]

La séquence finale obtenue après décomposition en ondelettes 1D de Haar est de même taille que la séquence initiale, et les coefficients correspondent au dernier coefficient de moyenne (soit S3) suivi des coefficients de détail des suites Dn du niveau le moins détaillé (n = 3) et niveau le plus détaillé (n = 1).
Donc : F = [5.375 1.125 –3.5 2.75 –1 –2 7 0.5]

Compression avec pertes

Transformée en ondelettes discrètes : **Exercice 2**

Soit l'image I suivant

$$I = \begin{bmatrix} 0 & 250 & 25 & 50 & 200 & 0 & 0 & 0 \\ 50 & 50 & 50 & 25 & 50 & 0 & 25 & 0 \\ 25 & 50 & 0 & 250 & 0 & 50 & 0 & 0 \\ 75 & 200 & 200 & 0 & 250 & 250 & 0 & 200 \\ 250 & 25 & 250 & 200 & 0 & 75 & 25 & 25 \\ 50 & 0 & 50 & 0 & 0 & 75 & 250 & 250 \\ 250 & 250 & 25 & 250 & 50 & 25 & 50 & 50 \\ 250 & 200 & 50 & 50 & 50 & 25 & 0 & 200 \end{bmatrix}$$

1. Donner la décomposition en ondelettes 1D de Haar de l'image I

2. Donner la matrice de quantification à l'aide d'une quantification uniforme à zone morte de largeur 30 et de pas de quantification 10.
3. Coder le résultat obtenu à l'aide de Huffman
4. Calculer le quotient, le taux et le gain de compression. Conclure

Transformée en ondelettes discrètes : Exercice 2 (solution)

Soit l'image I suivante

$$I = \begin{bmatrix} 0 & 250 & 25 & 50 & 200 & 0 & 0 & 0 \\ 50 & 50 & 50 & 25 & 50 & 0 & 25 & 0 \\ 25 & 50 & 0 & 250 & 0 & 50 & 0 & 0 \\ 75 & 200 & 200 & 0 & 250 & 250 & 0 & 200 \\ 250 & 25 & 250 & 200 & 0 & 75 & 25 & 25 \\ 50 & 0 & 50 & 0 & 0 & 75 & 250 & 250 \\ 250 & 250 & 25 & 250 & 50 & 25 & 50 & 50 \\ 250 & 200 & 50 & 50 & 50 & 25 & 0 & 200 \end{bmatrix}$$

1. Donner la décomposition en ondelettes 1D de Haar de l'image I

a. Suivant la largeur :

$$\left[\begin{array}{cccc|cccc} 125 & 37.5 & 100 & 0 & -125 & -12.5 & 100 & 0 \\ 50 & 37.5 & 25 & 12.5 & 0 & 12.5 & 25 & 12.5 \\ 37.5 & 125 & 25 & 0 & -12.5 & -125 & -25 & 0 \\ 137.5 & 100 & 250 & 100 & -62.5 & 100 & 0 & -100 \\ 137.5 & 225 & 37.5 & 25 & 112.5 & 25 & -37.5 & 0 \\ 25 & 25 & 37.5 & 25 & 25 & 25 & -37.5 & 0 \\ 250 & 137.5 & 37.5 & 50 & 0 & -112.5 & 12.5 & 0 \\ 225 & 50 & 37.5 & 100 & 25 & 0 & 12.5 & -100 \end{array}\right]$$

a. Suivant la hauteur:

$$\left[\begin{array}{cccc|cccc} -87.5 & 37.5 & 62.5 & 6.25 & -62.5 & 0 & 62.5 & 6.25 \\ -87.5 & 112.5 & 137.5 & 50 & -37.5 & -12.5 & -12.5 & -50 \\ -81.25 & 125 & 37.5 & 25 & 68.75 & 25 & -37.5 & 0 \\ -237.5 & 93.75 & 37.5 & 75 & 12.5 & -56.25 & 12.5 & -50 \\ \hline 37.5 & 0 & 62.5 & 6.25 & -62.5 & -12.5 & 37.5 & -6.25 \\ -50 & 12.5 & -112.5 & -50 & 25 & -112.5 & -12.5 & 50 \\ 56.25 & 100 & 0 & 0 & 43.75 & 0 & 0 & 0 \\ 12.5 & 43.75 & 0 & -25 & -12.5 & -56.25 & 0 & 50 \end{array}\right]$$

Compression avec pertes

Transformée en ondelettes discrètes : **Exercice 2 (solution)**

1. Donner la décomposition en ondelettes 1D de Haar de l'image I

−87.5	37.5	62.5	6.25	−62.5	0	62.5	6.25
−87.5	112.5	137.5	50	−37.5	−12.5	−12.5	−50
−81.25	125	37.5	25	68.75	25	−37.5	0
−237.5	93.75	37.5	75	12.5	−56.25	12.5	−50
37.5	0	62.5	6.25	−62.5	−12.5	37.5	−6.25
−50	12.5	−112.5	−50	25	−112.5	−12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	−25	−12.5	−56.25	0	50

a. Suivant la largeur:

62.5	34.375	25	28.125	−62.5	0	62.5	6.25
100	93.75	−12.5	43.75	−37.5	−12.5	−12.5	−50
103.125	31.25	−21.875	6.25	68.75	25	−37.5	0
165.625	56.25	71.875	−18.75	12.5	−56.25	12.5	−50
37.5	0	62.5	6.25	−62.5	−12.5	37.5	−6.25
−50	12.5	−112.5	−50	25	−112.5	−12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	−25	−12.5	−56.25	0	50

b. Suivant la hauteur:

81.25	64.0625	6.25	35.9375	−62.5	0	62.5	6.25
134.375	43.75	25	−6.25	−37.5	−12.5	−12.5	−50
−18.75	29.6875	18.75	−7.8125	68.75	25	−37.5	0
−31.25	−12.5	−46.875	12.5	12.5	−56.25	12.5	−50
37.5	0	62.5	6.25	−62.5	−12.5	37.5	−6.25
−50	12.5	−112.5	−50	25	−112.5	−12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	−25	−12.5	−56.25	0	50

Compression avec pertes

Transformée en ondelettes discrètes : **Exercice 2 (solution)**

1. Donner la décomposition en ondelettes 1D de Haar de l'image I

81.25	64.0625	6.25	35.9375	−62.5	0	62.5	6.25
134.375	43.75	25	−6.25	−37.5	−12.5	−12.5	−50
−18.75	29.6875	18.75	−7.8125	68.75	25	−37.5	0
−31.25	−12.5	−46.875	12.5	12.5	−56.25	12.5	−50
37.5	0	62.5	6.25	−62.5	−12.5	37.5	−6.25
−50	12.5	−112.5	−50	25	−112.5	−12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	−25	−12.5	−56.25	0	50

a. Suivant la largeur:

72.65625	8.59375	6.25	35.9375	−62.5	0	62.5	6.25
89.0625	45.3125	25	−6.25	−37.5	−12.5	−12.5	−50
−18.75	29.6875	18.75	−7.8125	68.75	25	−37.5	0
−31.25	−12.5	−46.875	12.5	12.5	−56.25	12.5	−50
37.5	0	62.5	6.25	−62.5	−12.5	37.5	−6.25
−50	12.5	−112.5	−50	25	−112.5	−12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	−25	−12.5	−56.25	0	50

b. Suivant la hauteur:

80.959375	26.953125	6.25	35.9375	−62.5	0	62.5	6.25
−8.103125	−18.359375	25	−6.25	−37.5	−12.5	−12.5	−50
−18.75	29.6875	18.75	−7.8125	68.75	25	−37.5	0
−31.25	−12.5	−46.875	12.5	12.5	−56.25	12.5	−50
37.5	0	62.5	6.25	−62.5	−12.5	37.5	−6.25
−50	12.5	−112.5	−50	25	−112.5	−12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	−25	−12.5	−56.25	0	50

Compression avec pertes

Transformée en ondelettes discrètes : Exercice 2 (Solution)

2. Donner la matrice de quantification à l'aide d'une quantification uniforme à zone morte de largeur z=30 et de pas de quantification Δ =10.

80.959375	26.953125	6.25	35.9375	−62.5	0	62.5	6.25
−8.103125	−18.359375	25	−6.25	−37.5	−12.5	−12.5	−50
−18.75	29.6875	18.75	−7.8125	68.75	25	−37.5	0
−31.25	−12.5	−46.875	12.5	12.5	−56.25	12.5	−50
37.5	0	62.5	6.25	−62.5	−12.5	37.5	−6.25
−50	12.5	−112.5	−50	25	−112.5	−12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	−25	−12.5	−56.25	0	50

L'application de l'équation de quantification ci-dessous **aux coefficients de détail** de la matrice précédente

- **Si | C | < Z, alors $C_{résultant} = 0$**
- **Si | C | ≥Z, alors $C_{résultant}$ = signe(C) × Δ × partie_entière (| C | / Δ)**

Avec

- **C : coefficient de détail**
- **Z: largeur de la zone morte**
- **Δ : le pas de quantification**

80.959375	0	0	30	−60	0	60	0
0	0	0	0	−30	0	0	−50
0	0	0	0	60	0	−30	0
−30	0	−40	0	0	−50	0	−50
30	0	60	0	−60	0	30	0
−50	0	−110	−50	0	0	0	50
50	100	0	0	40	0	0	0
0	40	0	0	0	−50	0	50

Transformée en ondelettes discrètes : Exercice 2 (Solution)

3. Coder le résultat obtenu à l'aide de l'arbre de Huffman

$$\begin{bmatrix} 80.959375 & 0 & 0 & 30 & -60 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0 & -30 & 0 & 0 & -50 \\ 0 & 0 & 0 & 0 & 60 & 0 & -30 & 0 \\ -30 & 0 & -40 & 0 & 0 & -50 & 0 & -50 \\ 30 & 0 & 60 & 0 & -60 & 0 & 30 & 0 \\ -50 & 0 & -110 & -50 & 0 & 0 & 0 & 50 \\ 50 & 100 & 0 & 0 & 40 & 0 & 0 & 0 \\ 0 & 40 & 0 & 0 & 0 & -50 & 0 & 50 \end{bmatrix}$$

Sur l'image quantifiée ci-dessus , *la compression HUFFMAN donne les fréquences d'apparition:*

Symboles	Nb Occurr.
0	38
-50	6
30	3
60	3

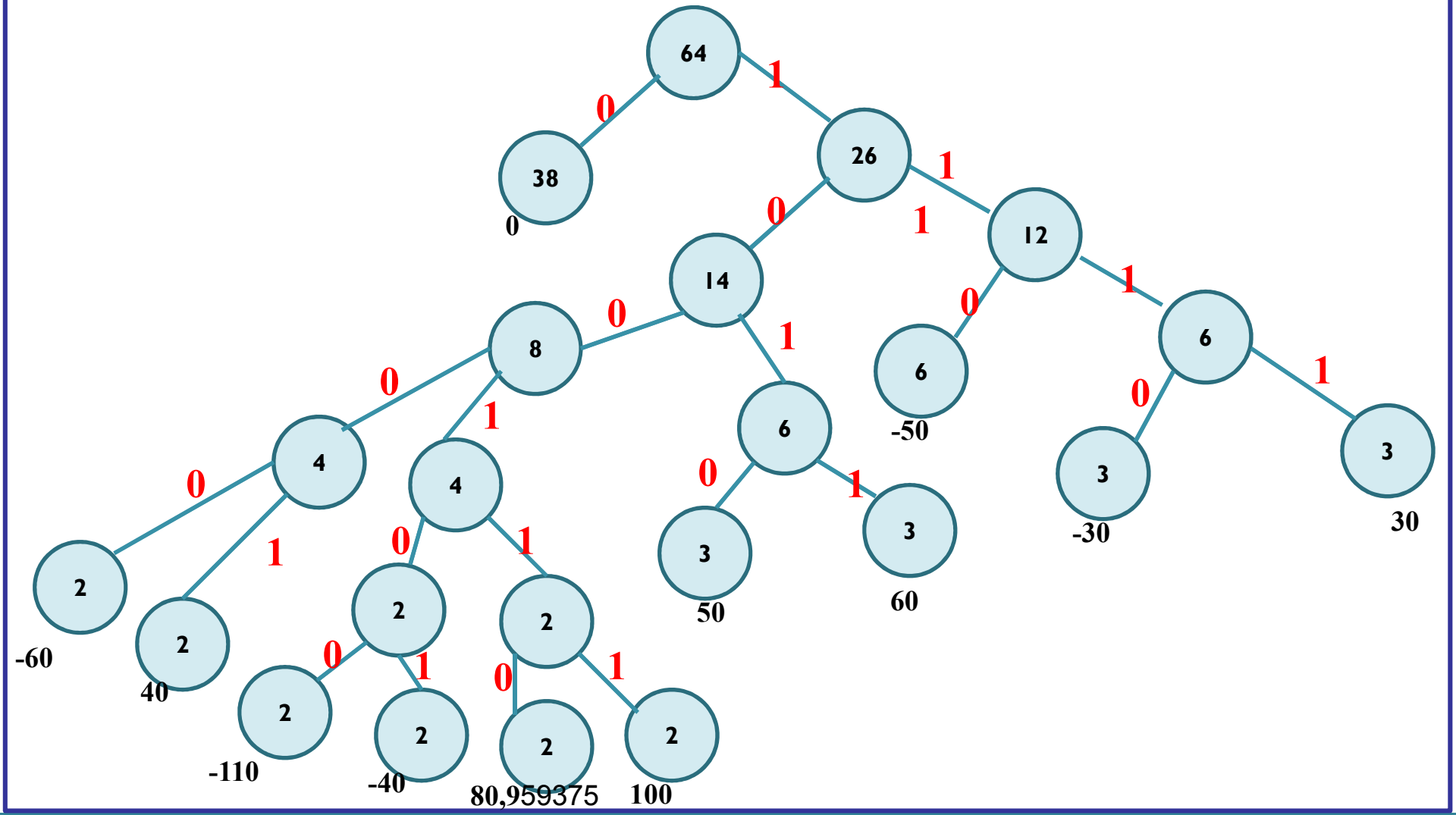
Symboles	Nb Occurr.
-30	3
50	3
-60	2
40	2

Symboles	Nb Occurr.
80.959375	1
-40	1
-110	1
100	1

Compression avec pertes

Transformée en ondelettes discrètes : **Exercice 2 (Solution)**

3. Coder le résultat obtenu à l'aide de l'arbre de Huffman



Transformée en ondelettes discrètes : Exercice 2 (Solution)

3. Coder le résultat obtenu à l'aide de l'arbre de Huffman

Symboles	Code binaire
0	0
-50	110
-30	1110
30	1111

Symboles	Code binaire
50	1010
60	1011
-60	10000
40	10001

Symboles	Code binaire
-110	100100
-40	100101
80,959375	100110
1000	100111

4. Calculer le quotient, le taux et le gain de compression. Conclure
   - La taille initiale de l'image est égale à

     $Taille\_initiale=L*H*poids_{pixel} = 8*8*8=512$ bits
   - La taille finale de l'image compressée est égale à :

     Taille_finale=(38*1+6*3+3*4+3*4+3*4+3*4+2*5+2*5+1*6+1*6+1*6+1*6)=148 bits

   - **quotient= Taille_initiale/ Taille_finale=3,4594**
   - **taux=1/ quotient=0,2890**
   - **gain =(1-taux)*100%= 71,1%**

Comparatif des méthodes de compression

- Sans compression
  - ✓Format très lourd.
  - ✓Exemples : BMP (pour des images d'une profondeur de pixel différente de 4 ou 8 bits).
- Compression sans perte
  - ✓RLE, LZW, Huffman : basé sur la redondance des valeurs.
  - ✓Exemples : PCX, GIF, PNG.
- Compression avec pertes
  - ✓Décomposition des images (détails / formes générales) et élimination de certaines parties de ces informations pour mieux compresser.
  - ✓Compression par DCT (JPEG)
    - Taux de compression réglable.
  - ✓Compression par ondelettes (JPEG2000)
    - Meilleur rapport compression/qualité d'image.
    - Compression sans pertes possibles (moins bonne compression).

Conclusion

A retenir pour ce cours:

Compression et stockage

- Position du problème
- Définition & Intérêts de la compression
- Types de compression
- Evaluation de la compression et des pertes

Compression avec pertes
DCT, Ondelettes

Compression sans perte
RLE, Huffman, LZW

Références

- **Livres**
  - ✓**Éric Incerti, *Compression d'image – Algorithmes et standards, Vuibert 2003***
  - ✓**Gilles Burel, *Introduction au traitement d'images – Simulation sous Matlab,* Hermès 2001 (chapitre 8)**

- **Sites web**
  - ✓**Cours de P. Nerzic (U. Rennes)**
    http://frama.link/K2vZFGkY
  - ✓**Basics of DCT and Entropy Coding, par Nimrod Peleg**
    www.lokminglui.com/J4DCT-Huff2009.pdf
  - ✓**Cours de D. Marshall (U. Cardiff)**
    http://www.cs.cf.ac.uk/Dave/Multimedia/PDF/ (*cf. chapitres 9,10 et 11)*