

Université Badji-Mokhtar. Annaba
Faculté de Technologie
Département d'Informatique



Chapitre 1

Techniques de base pour la compression

S3- 2025/2026



Introduction

“There was 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days, and the pace is increasing.”

-- Eric Schmidt, PDG Google, 2010



Introduction

- Exemple :

Source vidéo dont les caractéristiques:

- 25 images par seconde
- 16 millions de couleurs (3 octets par pixel)
- Résolution de 640 x 480
- Durée 2h
 - ✓ Taille sans compression = 165 Go, soit un débit de 23Mops.
 - ✓ Débit d'un lecteur CD-ROM simple 150 Kops(130 fois)
 - ✓ Taille d'un DVD est de 4.7 Go (34 DVD)

La compression est la
solution



Introduction

- **Le besoin de compression**
 - **Volume d'informations de plus en plus grand**
 - **Stockage: capacités de stockage limitées !!!**
 - **Transmission: bande passante des réseaux très chère**

Les solutions:

- **Augmenter les débits des réseaux (infrastructure)**
 - **Réduire la taille des données: compression**
- 



La compression

- La compression des données est une réduction du nombre de bits nécessaires pour représenter les données. Cela peut économiser la capacité de stockage, accélérer le transfert et réduire le coût du matériel de stockage et de la bande passante du réseau.
 - La méthode de compression dépend principalement du type de données à compresser.
 - La décompression effectue les opérations inverses pour reproduire les données originales.
- 

La compression

- Les critères de choix d'une méthodes de compression:

- Le quotient de compression

$$Q = \frac{\text{Taille du fichier initial}}{\text{Taille du fichier compressé}}$$

- Le taux de compression

$$T = \frac{\text{Taille du fichier compressé}}{\text{Taille du fichier initial}}$$

- le gain de compression

$$G = 1 - \frac{\text{Taille du fichier compressé}}{\text{Taille du fichier initial}}$$

- La qualité de compression: sans ou avec pertes, combien de pertes...

- La complexité des traitements



Types de compression

Compression symétrique ou asymétrique:

- **Compression Symétrique:** la même méthode est utilisée pour la compression ou la décompression: la même charge de calcul (e temps de calcul nécessaire pour la compression ou la décompression est équivalent).
 - ✓ C'est ce type de compression qui est généralement utilisée dans les transmissions de données.
 - ✓ **Exemples:** Lempel Ziv Walsh (LZW), Deflate (utilisé dans fichier zip, connexions HTTPs), etc.
- 



Types de compression

Compression symétrique ou asymétrique:

- **Compression Asymétrique:** complexités différentes entre compression et décompression selon l'application visée.
 - Exemples: LZWA (utilisé dans 7-Zip), MPEG, JPEG, etc.
- 



Types de compression

Compression statistique ou dictionnaire:

- **Compression statistique:** s'appuie sur l'analyse des données pour déterminer la fréquence à laquelle chaque symbole apparaît, puis attribue des codes binaires plus courts aux symboles les plus fréquents et des codes plus longs aux symboles les moins fréquents.
 - ✓ Utilisé dans les applications où les fréquences des symboles sont très différentes (JPEG, MPEG)
 - ✓ **Exemples:** Codage de Huffman, Codage de Shannon-Fano.
- 



Types de compression

Compression statistique ou dictionnaire:

- **Compression dictionnaire** : repose sur le remplacement de séquences de données répétées par des références à un dictionnaire, un ensemble de séquences déjà rencontrées.
 - ✓ Les données comportant beaucoup de répétitions: textes, fichiers exécutables, etc.
 - ✓ Exemple: Lempel ziv welch (LZW).
- 



Types de compression

Compression avec et sans pertes:

- **La méthode de compression sans pertes**, cela signifie que lorsque des données sont compressées et ensuite décompressées, l'information originale a été préservée. Aucune donnée n'a été perdue ou oubliée.
 - Objectif de compresser d'une séquence de données numériques sans perdre de l'information supprimer la redondance et garder l'information.
 - Utilisé pour des fichiers où la qualité est importante: .txt, csv, .PNG, .FLAC, .exe, .dll.
 - **Exemples:** Huffman, LZW
- 



Types de compression

Compression avec et sans pertes:

- **La méthode de compression avec pertes:** élimine certaines données pour réduire la taille total des données. Cela peut entrainer une dégradation de la qualité.
 - Les objectifs de la compression avec pertes sont d'éliminer les données non pertinentes.
 - Utilisé dans le domaine des fichiers multimédias: images, audio, vidéo.
 - **Exemples:** JPEG, MPEG, MP3.
- 

Types de compression

Compression sans pertes:

■ Notations :

- Source génère des séquences de données numériques (pixels , caractères, échantillons...) -> **symboles**

- Les symboles appartiennent à un ensemble fini appelé **alphabet**

$$A = \{a_1, a_2, \dots, a_i\} \text{ de taille } |A|$$

- $s = \{s_1, s_2, \dots, s_N\}$ la séquence générée par la source avec s_i de A pour tout i :
séquence source

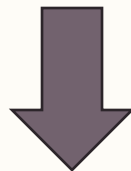
Types de compression

Compression sans pertes:

■ Exemple :

- Alphabet : $A = \{a, b, c, d\}$
 - Pour $N=8$ la séquence de symboles $s = (a, a, b, a, b, c, a, d)$
- ✓ La compression sans pertes doit définir une représentation de s avec un minimum de bits, et sans pertes d'information.

Comment calculer la quantité d'information ?



Théorie de l'information

La théorie de l'information

- Chaque **symbole** a_i de **l'alphabet** A à une probabilité d'apparition p_i
- La quantité d'information relative à ce symbole est

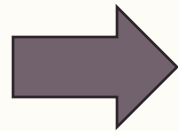
$$h_i = -\log_2(p_i) \text{ (Claude Shannon)}$$

- Si p_i est très petit, la quantité est très grande, alors que si p_i est très grand, elle est presque nulle.

La théorie de l'information

- **H(A)** est la quantité d'information moyenne contenu dans la source

L'entropie



$$H(A) = -\sum_{i=1}^{|A|} p_i \log_2(p_i)$$

- **Longueur minimale moyenne**

$$-N \sum_{i=1}^{|A|} p_i \log_2(p_i)$$

La théorie de l'information

■ Exemple 1:

- $A = \{a, b, c, d\}$ et $s = (a, a, b, a, b, c, a, d)$
- $p_a=0.5, p_b=0.25, p_c=0.125$, et $p_d=0.125$

Donc :

- $h_a=1, h_b=2, h_c=3$, et $h_d=3$
- $H(A)= 1.75$ bits par symbole
- Longueur minimale moyenne 14 bits



La théorie de l'information

- **Exemple 2:**
 - Prenons un texte de 2 mots, l'un ayant une fréquence de 500 et l'autre, de 250, pour un total de 750 ($n=750$).
 1. Calculer La quantité d'informations et l'entropie
 2. Combien bits pour stocker l'ensemble d'un texte?
- 

La théorie de l'information

Exercice:

- On considère les données contenues dans la structure suivante

1	2
3	4

- a) On suppose que les données 1, 2, 3 et 4 sont identiques.
- B) On suppose que les données $1 \neq 2 \neq 3 \neq 4$ sont différents entres elles.
- C) On suppose que les données $1 = 2 \neq 3 = 4$.
- Pour chaque cas calculer l'entropie et conclure sur le résultat trouvé.

Run-Length Encoding

- Run-Length Encoding (RLE) ou codage par longueurs de répétition
- Principe: Réduire la taille des chaînes répétitives (Run: nombre de répétitions)
- Algorithme:
 - Recherche des répétitions d'un caractère X
 - Changer la chaîne de répétitions par la paire (run,X)
- Exemple: 1- AAAAAAA -> 7A
 - 2- AAAABBBABBBBB -> 4A3B1A5B
- NB: pour les plages ≤ 2 il n'y a pas de compression



Run-Length Encoding

- **Exemple:** AAAAABCDDDD

- Taille avant compression (ASCII étendu) = ?

- Séquence compressée:

- Taille après compression =

- Taux de compression $T =$



Run-Length Encoding

- **Exemple:** AAAAABCDDDD

- Taille avant compression (ASCII étendu) = $11 * 8 = 88$ bits

- Séquence compressée: 5ABC4D

- Taille après compression = $6 * 8 = 48$ bits

- Taux de compression $T = 0.54$

Run-Length Encoding

- Utilisation: Images simples (icônes, logos), Les fichier fax, Text, Video (MPEG)
- Formats: BMP, PCX, TIFF, JPEG...
- Avantages:
 - ✓ Compression sans perte
 - ✓ Simple a implémenter et a comprendre
 - ✓ Exécution rapide
 - ✓ Efficace sur des données avec répétitions

Run-Length Encoding

- Inconvénients:
- ✓ Inefficacité sur les données non répétitives
- ✓ Exemple:

aabccdeefghijjjklmnopqrrstttuvwxyz

2a1b2c1d2e1f1g1h1i3j1k1l1m1o1p1q2r1s3t1u2v2w1x3y1z

négative compression

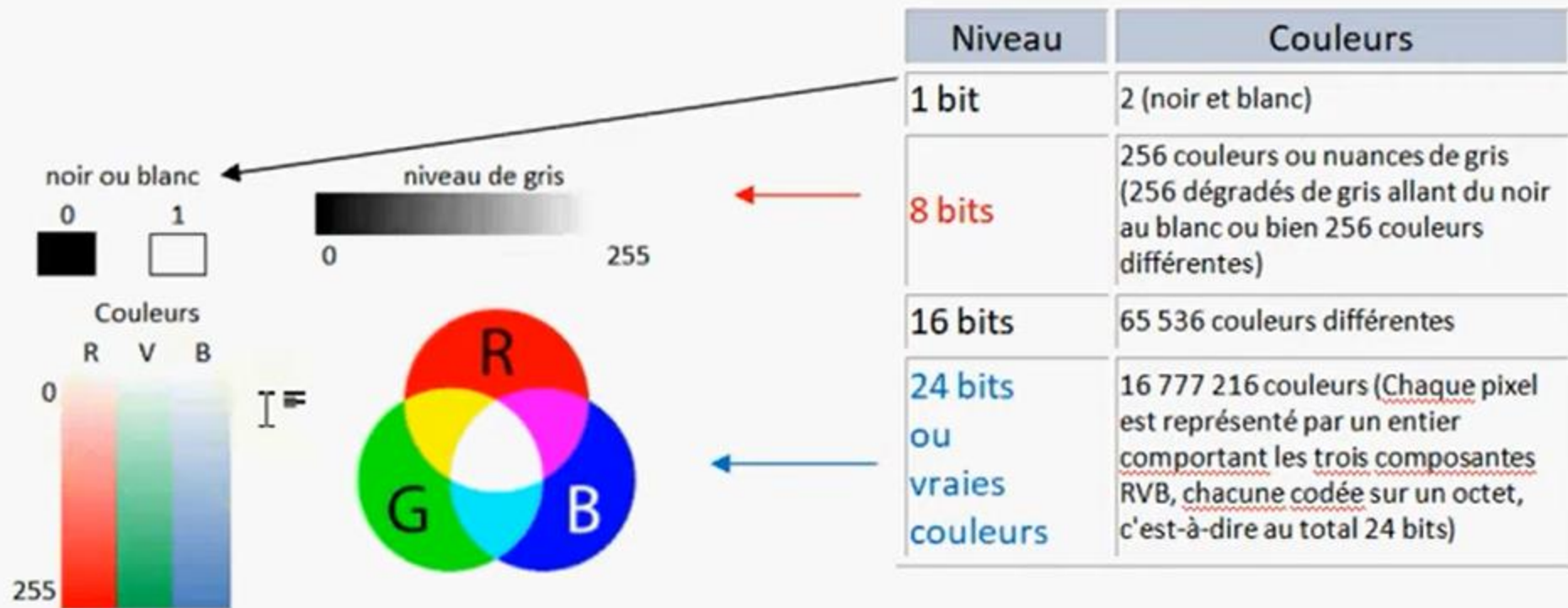


Run-Length Encoding

Compression RLE d'image

- Principe: Le RLE peut être utilisé pour compresser des images en niveaux de gris.
 - Chaque série de pixels de même intensité (niveau de gris) est codée sous forme de paire (**longueur de série, valeur de pixel**)
 - La longueur de série occupe généralement un octet.
 - La valeur de pixel occupe plusieurs bits, en fonction du nombre de niveaux de gris (généralement entre 4 et 8 bits).
- 

Run-Length Encoding



Run-Length Encoding

- **Exemple 1** : Une image bitmap en niveaux de gris profonds de 8 bits qui commence par: 12, 12, 12, 12, 12, 12, 12, 12, 12, 35, 76, 112, 67, 87, 87, 87, 5, 5, 5, 5, 5, 5, 1, ...

est compressé en: **9** ,12,35,76,112,67, **3** ,87, **6** ,5,1,.. . ., où les chiffres en gras indiquent les décomptes.

Run-Length Encoding

- **Exemple 2** : Une image bitmap en niveaux de gris profonds de 8 bits

Compression par RLE



134	134	134	27	27
27	27	0	0	0
0	0	239	239	239

Image non compressée

Run-Length Encoding

Comment parcourir l'image ?

