

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




# Chapitre 5

## Introduction à la Compression Vidéo Audio

S3- 2025/2026




# Introduction

- L'enregistrement **vidéo** et **audio** au format numérique implique d'équilibrer la qualité avec la taille du fichier et le débit (i.e. **un débit plus élevé offre une meilleure qualité, mais des fichiers plus volumineux**).
  - Donc, la plupart des formats utilisent la compression pour réduire la taille et le débit du fichier en réduisant sélectivement la qualité.
  - Les techniques de compression audio et vidéo **éliminent certains éléments d'information numériques inutiles ou redondants**, ou suppriment les informations numériques qui pourraient **ne pas être perceptibles par l'oreille ou l'œil humain**.
- **Compression avec pertes** (p. ex., MP3, AAC, AVC).
- 

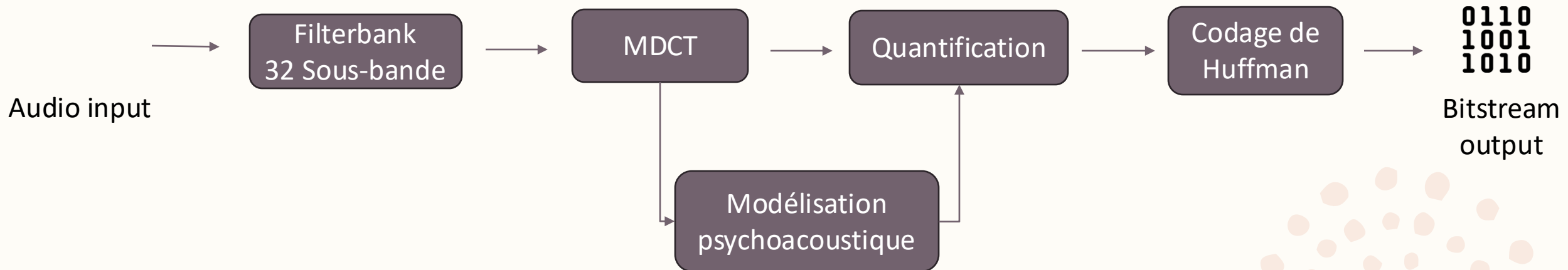


# Compression Audio

- Le **MP3** est un format de compression de **fichiers audio**, développé par le **Moving Pictures Experts Group** en 1994, intitulé **MPEG 1 - Audio Layer 3**.
  - La compression MP3 fait référence à la réduction de la taille du fichier d'une piste audio en **supprimant certaines de ses données**, donc compresse les fichiers audio **par perte de données (Compression destructrice du son)**.
  - L'oreille n'est pas sensible à toutes les fréquences audio: entre **20 Hz et 20 kHz**. Donc, il est possible de supprimer « **plus ou moins** » (en fréquences) les sons de **hautes et basses fréquences**.
- 

# Compression Audio

- La compression MP3 s'appuie sur les étapes suivantes :
- ✓ Diviser l'audio en trames (trame = 26 ms) et, pour chaque trame, en 32 sous-bandes.
- ✓ Transformer l'audio avec MDCT (Modified Discrete Cosine Transform)
- ✓ Modélisation psychoacoustique
- ✓ Quantification et allocation de bits
- ✓ Codage de Huffman



# Compression Audio

- **MDCT (Modified Discrete Cosine Transform):**

- ✓ Cette étape consiste à diviser l'audio en petites trames et à transformer les données du domaine temporel (forme d'onde sonore) en domaine fréquentiel (représentant différentes fréquences du son).
- ✓ La MDCT est une transformation de la forme suivante :

$$X_k = \sum_{n=0}^{N-1} x_n * \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right)$$

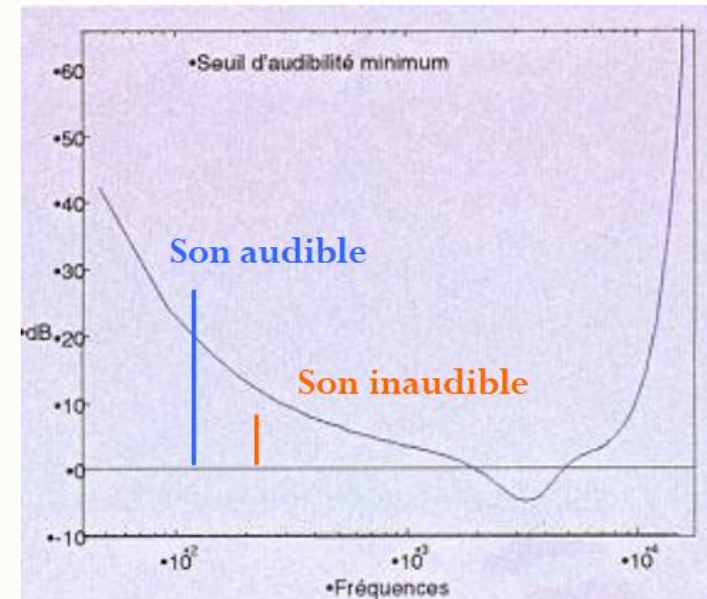
- $X_k$  est le k-ième coefficient de fréquence
- $x_n$  est le n-ième échantillon du domaine temporel
- $N$  est la taille du bloc de données audio (généralement  $N=576$  pour MP3)
- $k$  est l'indice de la fréquence.
- $n$  est l'indice temporel.

# Compression Audio

- **Modélisation psychoacoustique:**


- ✓ Sélectionne les parties du son qui sont moins susceptibles d'être entendues et qui peuvent donc être supprimées. Cela est basé sur :
- ✓ **Effets de masquage** : élimine les sons plus faibles et plus courts (**seuil de masquage**)
- ✓ **Sensibilité aux fréquences** : les humains sont plus sensibles aux fréquences moyennes (environ 500 Hz à 5 kHz) et moins sensibles aux fréquences très élevées (supérieures à 16 kHz) ou très basses.

La compression MP3 utilise cette fonctionnalité en réduisant les données audio des fréquences que nous sommes moins susceptibles de remarquer.





# Compression Audio

- **Quantification et allocation de bits:**
  - ✓ Le modèle psychoacoustique contrôle l'étape de quantification en comparant les coefficients de fréquence de MDCT aux seuils de masquage et détermine :
    - Quel coefficient conserver?
    - Quel coefficient quantifier de manière intensive ?
    - Quel coefficient éliminer ?
    - Combien de bits utiliser par sous-bande?
- 

# Compression Audio

- **Quantification et allocation de bits:**

✓ Pour chaque coefficient MDCT  $x$ , MP3 applique cette équation:

$$q = \text{round}\left(\frac{|x|^{3/4}}{2^{global\_gain/4}}\right) \cdot \text{sign}(x)$$

- $x$ : coefficient MDCT
- $Global\_gain = 4 * \log_2(\max(|x|^{3/4}))$
- $Sign(x)$ : le signe original + ou –
- $3/4$ : est choisi pour correspondre à la perception du volume sonore par l'oreille.


- **Déquantification:**

$$\hat{x} = \text{sign}(x) \cdot |q|^{4/3} \cdot 2^{global\_gain/4}$$





# Compression Audio

- **Quantification et allocation de bits:**
  - ✓ Après la quantification, le coefficient important est conservé (haute précision), et les coefficients quantifiés à 0 sont complètement supprimés.
  - ✓ **Allocation de bits** : les bits disponibles sont alloués entre les coefficients de fréquence. Les composants les plus importants (en fonction de leur volume, de leur masquage et de leur sensibilité) se voient attribuer davantage de bits, tandis que les composants moins importants reçoivent moins de bits. Cette répartition permet d'équilibrer la taille du fichier et la qualité audio.
- 

# Compression Audio


- **Codage de Huffman:**
- ✓ MP3 utilise des tables prédéfinies : **32 tables de Huffman**.
- ✓ Pour chaque trame, on **sélectionne une table de Huffman** qui correspond le mieux au coefficient quantifié.
- ✓ Chaque **coefficient quantifié** est ensuite remplacé par le **code à longueur variable** correspondant dans la table choisie.
- ✓ Le signe d'une valeur non nulle est ajouté comme un **bit supplémentaire** après le code de Huffman.
- ✓ **Avantages:** Évitez de créer un arbre de Huffman pour chaque trame, et n'importe quel décodeur pourra décoder bitstreams à l'aide des tables standard.

# Compression Vidéo

- **La compression vidéo utilise trois techniques principales :**
- **Compression temporelle** (entre les images): elle supprime les informations répétées entre les images consécutives. Au lieu de stocker chaque image complète, la compression temporelle ne stocke qu'une seule image complète et n'enregistre que les changements pour les images suivantes.
- **Compression spatiale** (à l'intérieur de l'image): également appelée **compression intra-image**, elle supprime les détails ne sont pas facilement visibles à l'œil humain. Elle utilise des méthodes telles que la DCT ou la DWT, et la quantification.
- **Codage entropique**: les données sont encodées avec Huffman ou CABAC/CAVLC (dans H.264).

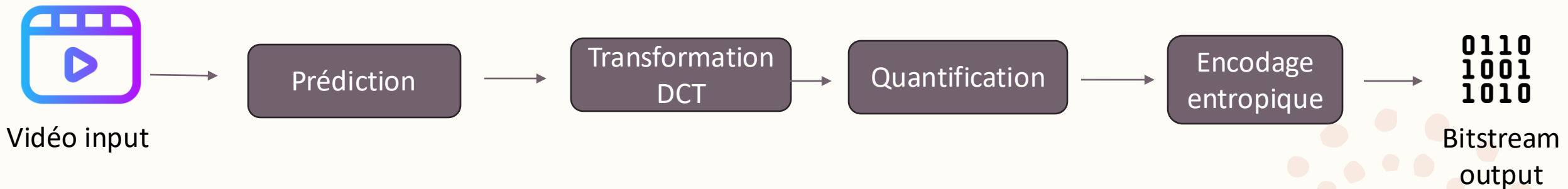


# Compression Vidéo

- **MP4** ou **MPEG-4 PART 14 (2003)** consiste à réduire la taille du fichier vidéo stocké au format MP4, qui est couramment utilisé pour le **stockage de vidéos et d'audio numériques**.
  - **MP4** est un format multimédia populaire qui prend en charge la vidéo, l'audio, les sous-titres et les images fixes.
  - La compression dans **MP4** vise à réduire les données nécessaires pour représenter le contenu vidéo et audio tout en maintenant la qualité.
  - La norme de **compression vidéo** la plus courante utilisée dans les fichiers MP4 est **H.264** (ou AVC, Advanced Video Coding). Pour l'**audio**, **AAC** (Advanced Audio Codec) est largement adopté dans **MP4**.
- 

# Compression Vidéo

- **H.264: or Advanced Video Coding (AVC):** est une norme de compression vidéo permettant de réduire la taille (de 50 à 60 %) des vidéos tout en conservant une bonne qualité.
- H.264 est largement utilisée pour le format MP4, le streaming, YouTube, la visioconférence, caméras de téléphone, etc.
- La compression H.264 repose sur les étapes suivantes :



# Compression Vidéo

- **Prédiction:**

- Le H.264 divise les images en 3 types principaux :
- **Image I (I-Frame)** : Une image est une image I si aucune image précédente ne permet de prédire l'image actuelle (coupure de scène). L'image I est l'image complète stockée comme une image JPEG.
- **Image P (P-Frame)**: Une image est une image P si elle est similaire à l'image précédente. L'image P stocke uniquement les changements et les mouvements des objets.
- **Image B (B-Frame)**: Une image est une image B si elle est similaire à la fois à l'image précédente et à l'image suivante. L'image B stocke uniquement les changements et les mouvements des objets.

# Compression Vidéo

## ■ Prédiction:

- La prédiction permet de supprimer la redondance au sein d'une même image (prédiction spatiale) ou entre les images (prédiction temporelle).
- Étapes de la prédiction :
  1. Chaque image (I, P ou B) est divisée en blocs (macroblochs) de 16x16 pixels.
  2. **Prédiction intra (I-Frame):** Prédire les valeurs des pixels en utilisant les pixels des blocs voisins ayant des directions communes : **Top, Left, TopLeft, diagonale**, etc. Calculer ensuite **la valeur résiduelle = bloc d'origine – bloc prédit**.




# Compression Vidéo

## ■ Prédiction:

2. **Prédiction inter (P-Frame):** Prédiction à partir de l'image de référence précédente (I ou P). Calculer ensuite **la valeur résiduelle = bloc d'origine – bloc prédit**. P stocke le mouvement et les valeurs résiduelles.


3. **Prédiction bidirectionnelle (B-Frame):** Prédiction basée sur les images précédentes et suivantes (moyenne de la valeur précédente et de la valeur suivante). Ensuite calculer **la valeur résiduelle = bloc d'origine – prédiction bidirectionnelle**. B stocke le mouvement et les valeurs résiduelles.







# Compression Vidéo

- **Transformation DCT** : convertit le bloc du domaine spatial au domaine fréquentiel.
  - **Quantification**: La quantification est une compression avec perte dans laquelle de nombreux coefficients deviennent nuls après la quantification. H.264 utilise un paramètre de quantification (0-51) :
    - ✓ **QP élevé** : compression plus importante, qualité moindre.
    - ✓ **QP faible** : compression plus faible, qualité supérieure.
- 

# Compression Vidéo

- **Encodage entropique: CAVLC: Context Adaptive Variable Length Coding**
  - Exemple : [3,1,0,0,0,0,0,0,0, 1, 0,0,0,0,0,0]
  - Compter les coefficients non nuls: [3,1,1] = 3 et Trailing one: dernier coefficient = 1
  - Ces valeurs sont encodées à l'aide de la table VLC.
  - CAVLC stocke le bit de signe (Trailing one): 0 (positif).
  - CAVLC encode les coefficients restants à l'aide de différentes tables VLC.
  - CAVLC compte tous les zéros précédant le dernier coefficient non nul. Ensuite, il encode le nombre de zéros précédant chaque coefficient non nul à l'aide des tables VLC (runbefore).

# Compression Vidéo

- **Encodage entropique: CAVLC: Context Adaptive Variable Length Coding**
  - Exemple : [3,1,0,0,0,0,0,0,0, 1, 0,0,0,0,0,0]
  - Compter les coefficients non nuls: [3,1,1] = 3 et Trailing one: dernier coefficient = 1
  - Ces valeurs sont encodées à l'aide de la table VLC.
  - CAVLC stocke le bit de signe (Trailing one): 0 (positif).
  - CAVLC encode les coefficients restants à l'aide de différentes tables VLC.
  - CAVLC compte tous les zéros précédant le dernier coefficient non nul. Ensuite, il encode le nombre de zéros précédant chaque coefficient non nul à l'aide des tables VLC (runbefore).



# Compression Vidéo

- **ACC (Advanced Audio Coding):** est une norme de compression audio avec perte développée par le groupe MPEG. AAC offre une qualité sonore nettement supérieure à débit binaire égal et des performances optimales à faible débit.
  - AAC est largement utilisée pour le format MP4, le streaming, YouTube, iTunes, les smartphones, TV, les plateformes vidéo en ligne, etc.
  - Il utilise des outils avancés tels que les fenêtres MDCT longues et courtes, une modélisation psychoacoustique améliorée, la mise en forme temporelle du bruit (TNS), un codage stéréo flexible et un codage entropique plus efficace.
- 