

# Signaux, Sons et Images pour l'Informaticien: Compression de sons

Diane Lingrand

Polytech SI3

2016 - 2017

# Outline

- Généralités
- Algorithmes de codage : Rice, RLC, Huffman
- FLAC
- MP3, Ogg Vorbis, WMA, AAC

- Compression avec / sans pertes
- Taux de compression - Débit binaire
- Que cherche t-on ?
  - écouter : sur quel matériel ?
  - reconnaître / transcrire
  - archivage



Propriétés des CDs :

- Fréquence d'échantillonnage : 44.1 kHz
- Encodage des valeurs : 16 bits = 2 octets
- Stéréo : 2 canaux

Pas de compression :

- 1 seconde prend  $2 * 2 * 44.1 * 1000 = 172.3kO$
- 1 minute prend  $60 \times 172.3kO = 10.1Mo$
- 1 CD (650 Mo) contient 64 minutes
- débit binaire : 1411,2 kbps

# Codage de Rice

- Codage entropique de paramètre  $k$
- Équivalent au codage de Golomb de paramètre  $2^k$
- Principe du codage d'un entier  $N$  :
  - codage unaire de  $q = \frac{N}{2^k}$
  - codage binaire de  $r = N - 2^k \times q$
- longueur du code :  $E(N/2^k) + 1 + k$
- Un exemple : coder  $27 = 11011$

$$k = 2 : 111111011$$

<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>
quotient			reste	
en unaire	+0	en binaire		
<b>111111</b>	<b>0</b>	<b>1</b>	<b>1</b>	

$$k = 3 : 1110011$$

<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>
quotient			reste	
en unaire	+0	en binaire		
<b>111</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>

# Codage de Rice : quelques exemples

déc.	binaire	Rice ( $k = 0$ )	Rice ( $k = 1$ )	Rice ( $k = 2$ )	Rice ( $k = 3$ )	Rice ( $k = 4$ )
0	0000	0	0 0	0 00	0 000	0 0000
1	0001	10	0 1	0 01	0 001	0 0001
2	0010	110	10 0	0 10	0 010	0 0010
3	0011	1110	10 1	0 11	0 011	0 0011
4	0100	11110	110 0	10 00	0 100	0 0100
5	0101	111110	110 1	10 01	0 101	0 0101
6	0110	1111110	1110 0	10 10	0 110	0 0110
7	0111	11111110	1110 1	10 11	0 111	0 0111
8	1000	111111110	11110 0	110 00	10 000	0 1000
9	1001	1111111110	11110 1	110 01	10 001	0 1001
10	1010	11111111110	111110 0	110 10	10 010	0 1010

- Principe : compter le nombre d'occurrences des valeurs.
- Efficace si des valeurs se répètent.
- Exemple : AABBCCCCBBBBBBA codé en 2A2B4C6B1A
- Exemple : ABCBAC codé en 1Q1B1C1B1A1C
- A été utilisé à la télévision, en images avant GIF

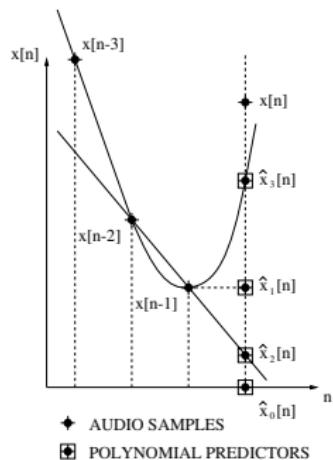
<http://xiph.org/flac/format.html>

Compression sans pertes en 4 étapes :

- découpage en blocs (environ 13 à 26 ms)
- décorrélation entre canaux : moyenne et différence pour la stéréo
- prédiction : 4 méthodes
  - verbatim : prédiction à zéro. Le signal est donc dans le résidu.
  - constant : pour le silence ou valeur constante. Utilise ensuite le RLC ou codage par plages
  - FLP (Fixed Linear Predictor) : on ne stocke que l'ordre du prédicteur.
  - FIR (Finite Impulse Response) linear prediction : nécessite de stocker les coefficients.
- codage des résidus : codage de Rice avec 1 coefficient par partition des données

# FLAC : prédicteur à coefficients entiers fixes

Le degré de prédicteur correspond également à la latence ou retard au démarrage.

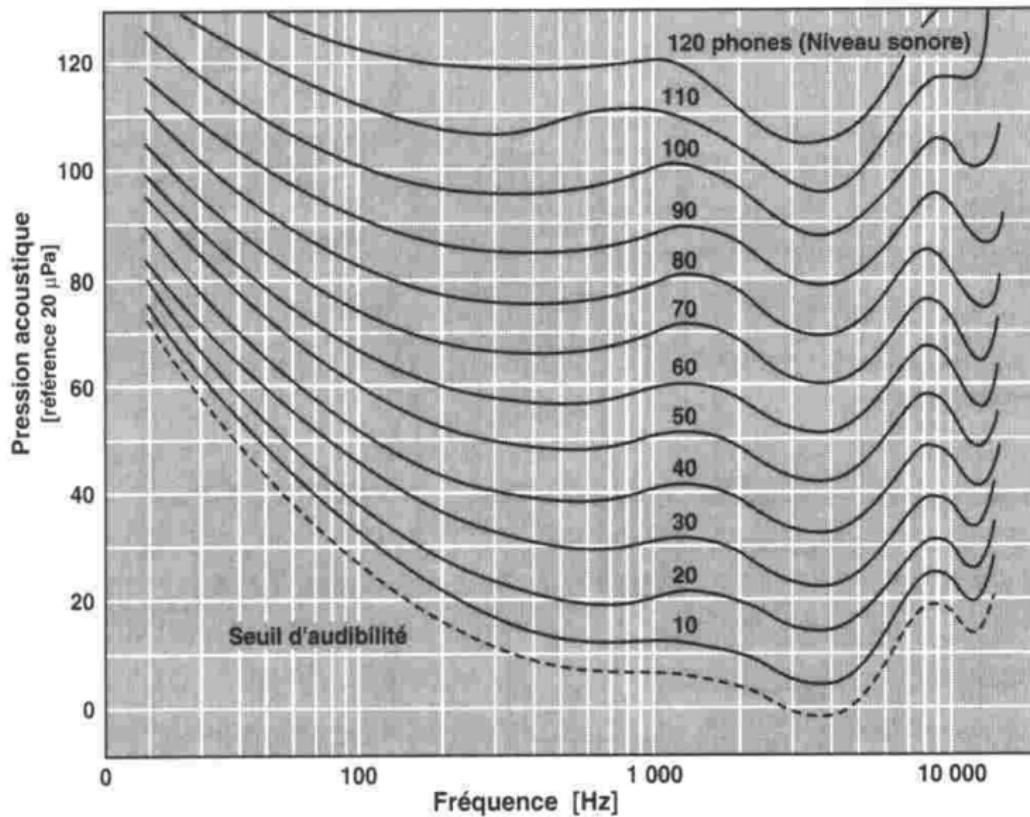


crédit : *Lossless Compression of  
Digital Audio, Mat Hans, Ronald W.  
Schafer*

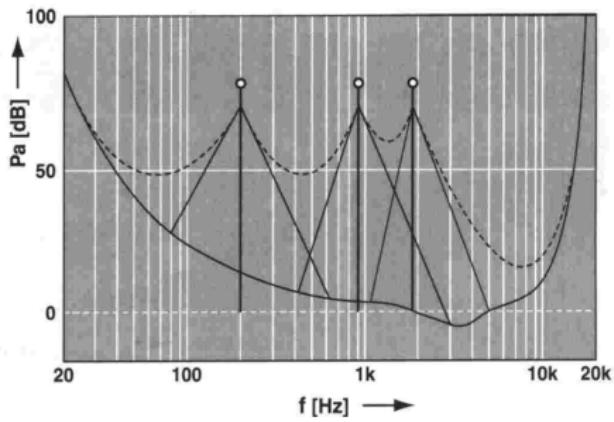
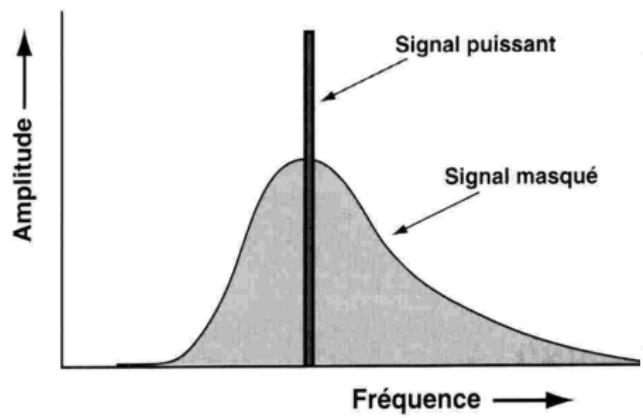
$$\begin{aligned}0 &: x[n] = 0 \\1 &: x[n] = x[n - 1] \\2 &: x[n] = 2x[n - 1] - x[n - 2] \\3 &: x[n] = 3x[n - 1] - 3x[n - 2] + x[n - 3] \\4 &: x[n] = 4x[n - 1] - 6x[n - 2] + 4x[n - 3] - x[n - 4]\end{aligned}$$

- Brevets. Spécifications libres de droit.
- Compression avec pertes
- Codage perceptuel : exploitation d'un modèle psycho-acoustique
- Débit variable (adaptation du débit à la quantité d'informations)

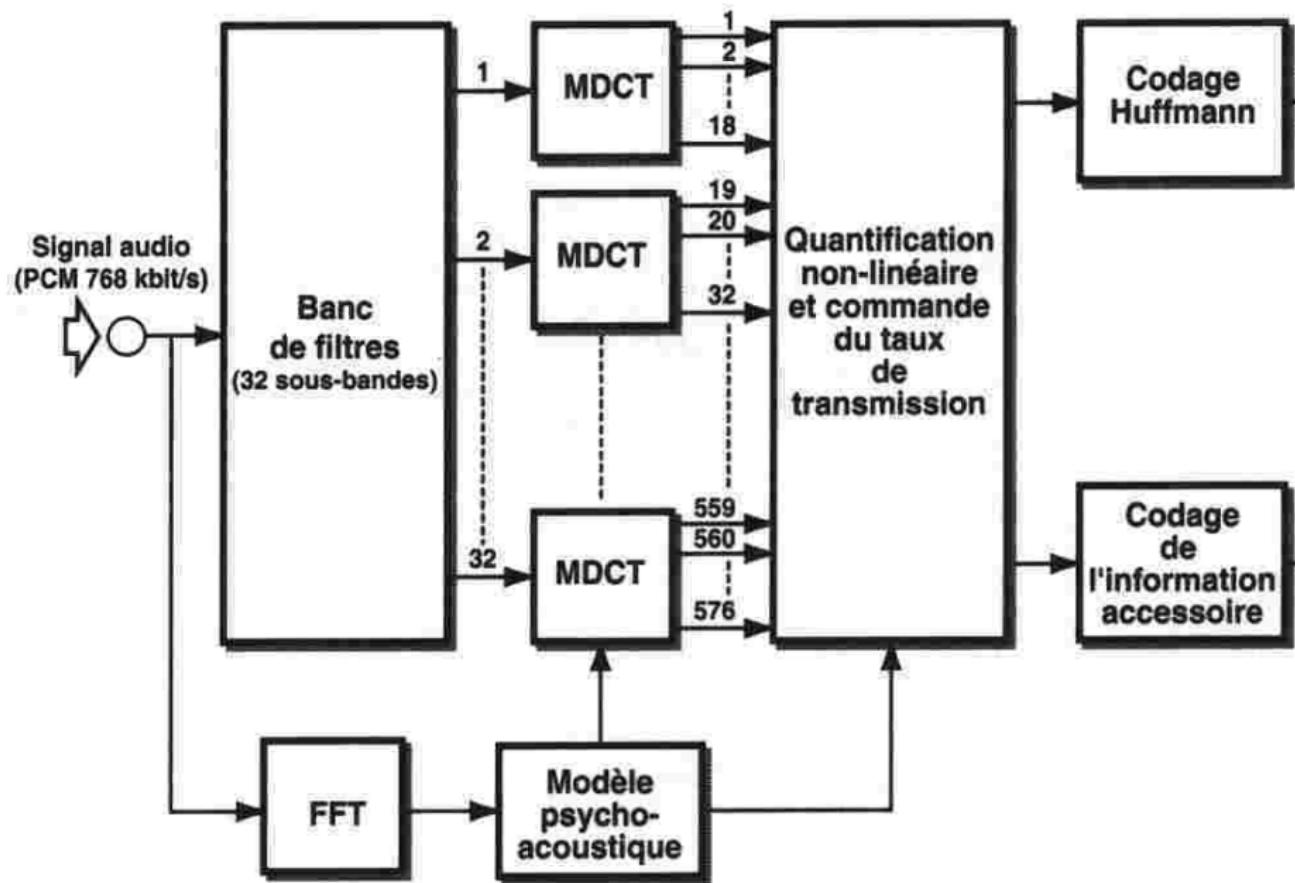
# Perception des sons



# Effet de masquage



# Compression MPEG-1 Audio Layer 3



- codage entropique (longueur variable)
- code préfixe (aucun code n'est préfixé par un autre code)
- 2 phases :
  - phase descendante : construction de l'arbre
  - phase ascendante : codage de l'information

## Codage d'Huffman : construction de l'arbre

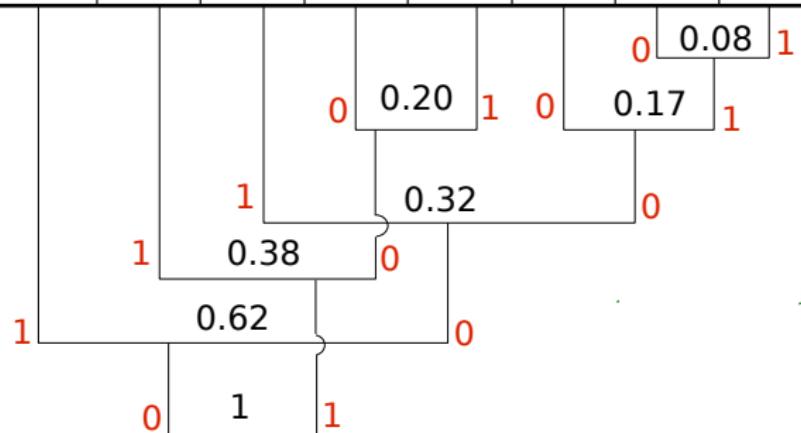
valeur $n_i$	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$
probabilités	0.3	0.18	0.15	0.10	0.10	0.09	0.05	0.03

- 1ère étape : passage de  $m=8$  à  $m=7$  :  $n_7$  et  $n_8$  ont les probabilités les plus faibles : on les regroupe en un élément  $n_{7,8}$  de probabilité 0.08

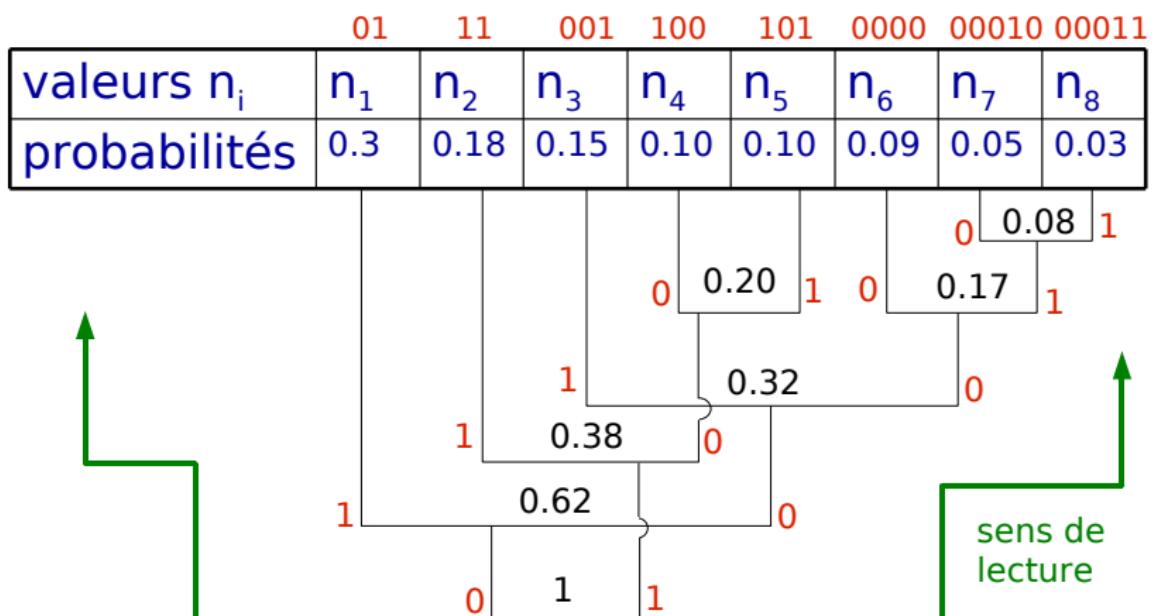
valeur $n_i$	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_{7,8}$
probabilités	0.3	0.18	0.15	0.10	0.10	0.09	0.08

# Codage d'Huffman : construction de l'arbre

valeurs $n_i$	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$
probabilités	0.3	0.18	0.15	0.10	0.10	0.09	0.05	0.03



# Codage d'Huffman : codage de l'information



- Codage d'un mot
  - codage ordinaire : sommes des  $p(n_i) * 3$  bits
  - codage d'Huffman : somme des  $p(n_i) * l_i$  bits
  - pour notre exemple : 3 bits contre 2.79 bits
  - pour une image (640\*480) : économie de 64512 bits=8kO
- Codage de l'arbre
  - statique : pas besoin de coder l'arbre
  - semi-statique : arbre calculé 1 fois pour toutes les données
  - adaptatif : modifications d'un arbre non transmis

- Projet Open Source. 1993 (standardisation ISO du MP3 en 1992)
- fréquence d'échantillonnage de 8 kHz à 48 kHz, quantification sur 16 bits ou +, polyphonie
- Encodeur en 3 phases :
  - MDCT, recouvrement, et fenêtrage (2 tailles de fenêtres)
  - quantification selon modèle psycho-acoustique
  - codage d'Huffman et bitpacking
- bit rate constant ou variable

- brevet déposé par Microsoft en 1999.
- bitrate constant
- pour une qualité similaire, meilleur taux de compression que mp3 (96kbps contre 128 kbps)
- algorithme similaire au mp3 mais meilleure connaissance/exploitation de la psycho-acoustique

- codec audio basé sur la norme Mpeg4, d'où son surnom MP4.
- extension des fichiers : .m4a
- algorithme similaire au mp3
- plusieurs canaux

# Compressions

taille (kO)	codage	taille (kO)	codage
3364	original		
1772	FLAC rapide	72	WMA 24 kbps
1640	FLAC lent	116	WMA 40 kbps
336	MP3	224	WMA 80 kbps
428	MP3	332	WMA 128 kbps
604	MP3	1320	WMA 320 kbps
768	MP3	140	Ogg Vorbis Q0
1044	AAC 10	352	Ogg Vorbis Q5
1060	AAC 100	1100	Ogg Vorbis Q10