# Audio Encoding and Modification

Mathieu Lagrange

February 5, 2019

Course website: https://github.com/mathieulagrange/datasim

# Outline

1. Audio coding

2. Time Scale Modification

# Outline

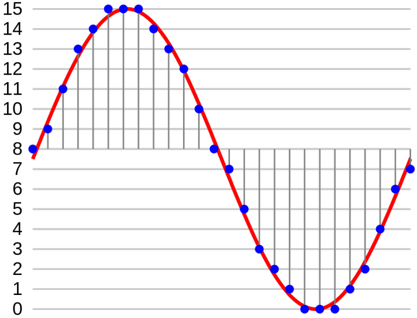
1. Audio coding

2. Time Scale Modification

# Types of encoding

- Pulse Code Modulation (PCM): WAV, AIFF, ...
- lossless encoding: FLAC, Apple Lossless
- lossy encoding: MP3, AAC, WMA, Vorbis

# Pulse code modulation




- Sampling frequency: 44.1 kHz
- Sample resolution: 16 bits
- Bit rate: $\approx$ 700 kbit/s

# Lossless encoding

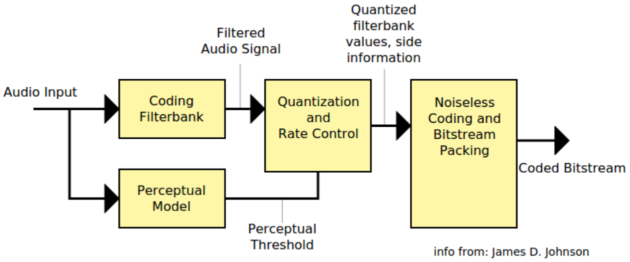
Free Lossless Audio Coding (FLAC)

- PCM -> FLAC -> PCM
- compression rate: $\approx$ 55 % (ZIP PCM $\approx$ 15)
- efficient streaming and decoding scheme

# Lossy Encoding

- PCM -> lossy -> PCM′
- balance bitrate reduction / perceptual distortion

# Block diagram of lossy audio coders



# A bit of signal processing

Need for a representation that:

- project the input signal over the set of audible frequencies
- invertible
- bit wise efficient

# Spectral representation

- Short Term Fourier Transform (STFT)
- Discrete Cosine Transform (DCT)
- Modified Discrete Cosine Transform (MDCT)

# Spectral representation

- Short Term Fourier Transform (STFT)

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{\frac{-i2\pi kn}{N}}$$

with

$$0 \leq k < N \text{ and } X_k \in \mathbb{C}$$

- Discrete Cosine Transform (DCT)
- Modified Discrete Cosine Transform (MDCT)

# Spectral representation

- Short Term Fourier Transform (STFT)
- Discrete Cosine Transform (DCT)

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

with

$$0 \leq k < N \text{ and } X_k \in \mathbb{R}$$

- Modified Discrete Cosine Transform (MDCT)

# Spectral representation

- Short Term Fourier Transform (STFT)
- Discrete Cosine Transform (DCT)
- Modified Discrete Cosine Transform (MDCT)

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

with

$$0 \leq k < N \text{ and } X_k \in \mathbb{R}$$

# The uncertainty principle (Heisenberg)

The uncertainty principle states that there is a fundamental limit to the precision with which certain pairs of complementary variables can be known simultaneously.
In quantum mechanics, the variables of interest are the position $x$ and momentum $p$, and:

$$\sigma_x \sigma_p \geq \frac{\hbar}{2}$$

# The uncertainty principle (Heisenberg)

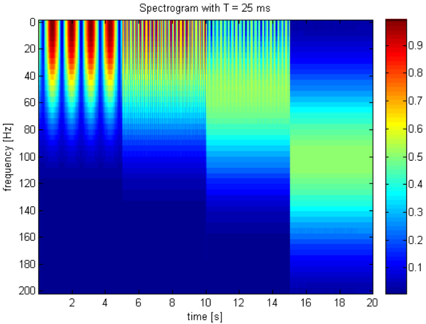
The uncertainty principle states that there is a fundamental limit to the precision with which certain pairs of complementary variables can be known simultaneously.
In signal processing, the variables of interest are the time $t$ and frequency $F$.

Example:

$$x(t) = \begin{cases} \cos(2\pi 10t); & 0 \leq t < 5s \\ \cos(2\pi 25t); & 5 \leq t < 10s \\ \cos(2\pi 50t); & 10 \leq t < 15s \\ \cos(2\pi 100t); & 15 \leq t < 20s \end{cases}$$

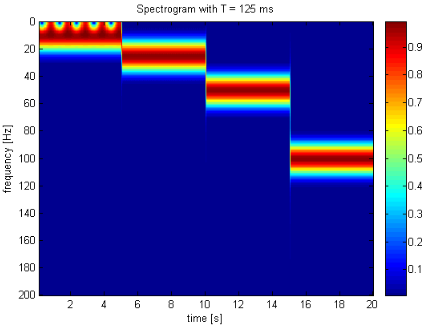
# The uncertainty principle (Heisenberg)

The uncertainty principle states that there is a fundamental limit to the precision with which certain pairs of complementary variables can be known simultaneously.



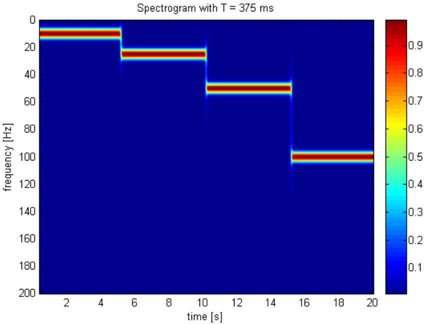
# The uncertainty principle (Heisenberg)

The uncertainty principle states that there is a fundamental limit to the precision with which certain pairs of complementary variables can be known simultaneously.



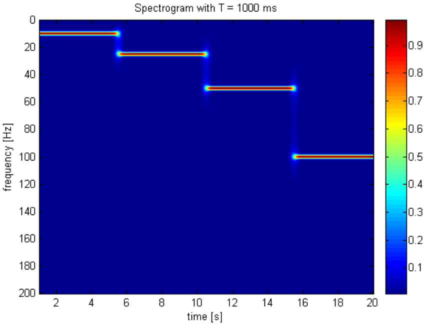
# The uncertainty principle (Heisenberg)

The uncertainty principle states that there is a fundamental limit to the precision with which certain pairs of complementary variables can be known simultaneously.



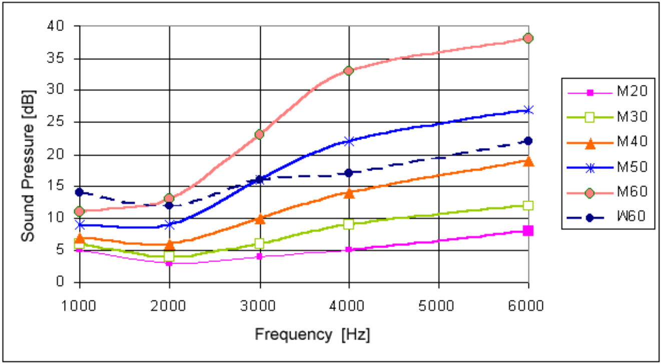
# The uncertainty principle (Heisenberg)

The uncertainty principle states that there is a fundamental limit to the precision with which certain pairs of complementary variables can be known simultaneously.



# A bit of psychoacoustic

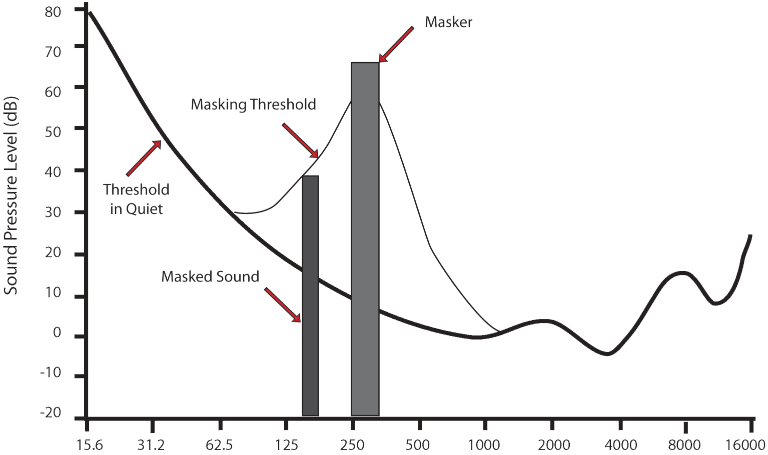
- Absolute threshold of hearing



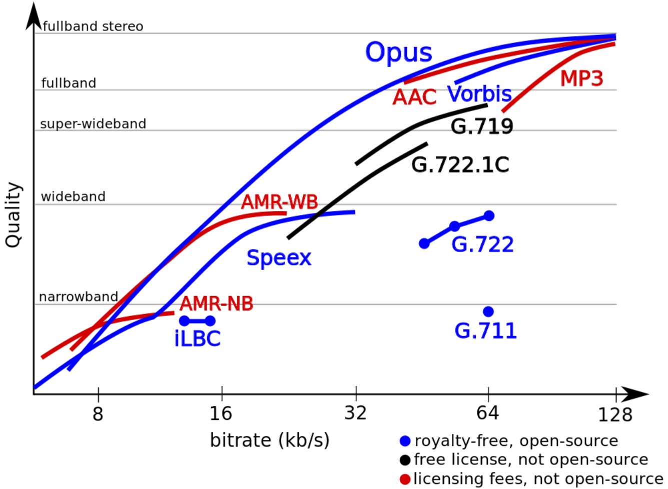

- Frequency masking

# A bit of psychoacoustic

- Absolute threshold of hearing
- Frequency masking



# A panorama


fullband stereo
fullband
super-wideband
wideband
narrowband
Quality
Opus
AAC
Vorbis
MP3
G.719
G.722.1C
AMR-WB
Speex
G.722
AMR-NB
iLBC
G.711
8
16
32
64
128
bitrate (kb/s)
royalty-free, open-source
free license, not open-source
licensing fees, not open-source

# MPEG-2 Layer III (MP3)

Life of a standard

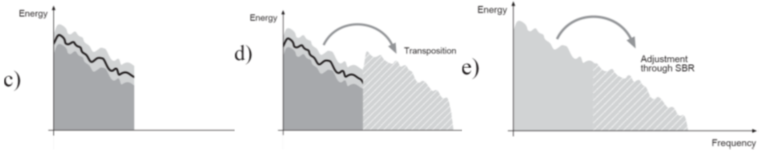
- Suzanne Vega: the mother of MP3
- 1994: Fraunhofer (Karlheinz Brandenburg) releases WinPlay3
- 1997: Winamp
- 1999: Napster

# Advanced Audio Coding (AAC)

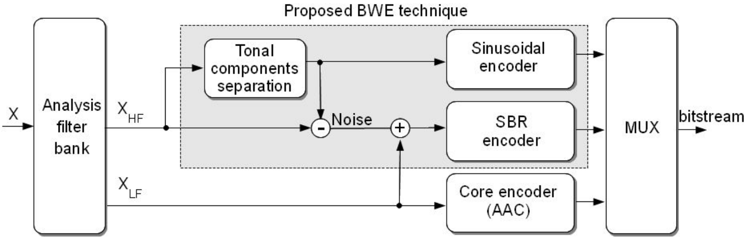
Replacement of MP3 (end of story ?)

- pure MDCT
- Higher coding efficiency for stationary signals: blocksize of 1024 or 960 samples
- Higher coding accuracy for transient signals: blocksize of 128 or 120 samples
- large set of tools to increase compression efficiency (SBR, TNS, ...)

# Spectral Band Replication (SBR)



# Spectral Band Replication (SBR)

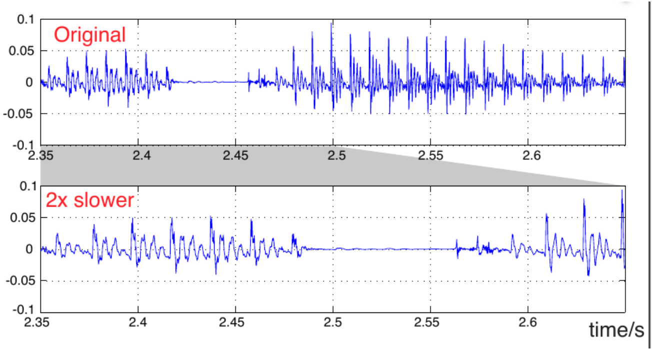


# Time Scale Modification (TSM)

Why do we want to make sound "quicker or slower" ?

- examine details in speech / performance
- to adjust durations
- to synchronize tracks
- to modify pitch

# Sampling rate

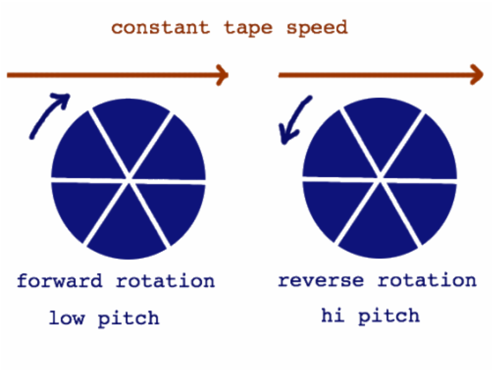



We can adjust the sampling rate: $x_s(t) = x(t/r)$

# Time & Pitch

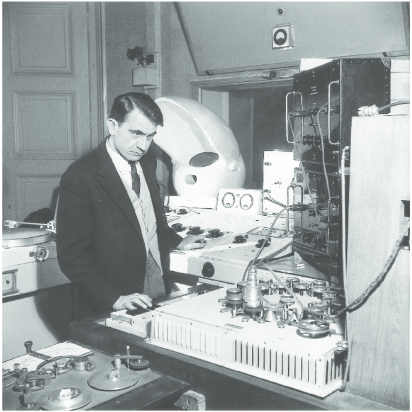
- Changing the sampling rate alers time and pitch
- Preserve pitch, change time: keep local time structure but changing global time course
- Preserve time, change pitch: analogous problem

# Analog TSM



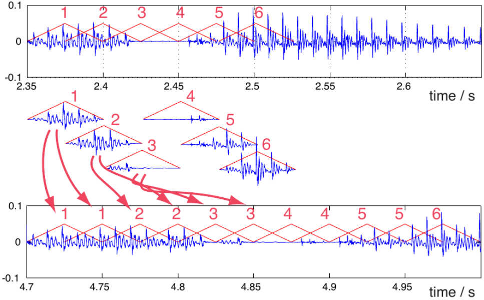

Magic is achieved trough rotating tape heads

# Ancestor: the phonogene



Pierre Schafer in front of the "phonogene" (1963) at the "Groupe de Recherches Musicales" (GRM)

# Digital equivalent




The Overlap and Add technique:

$$y^m[mL+n] = y^{m-1}[mL+n] + w[n]x[\lceil \frac{m}{r} \rceil L + n]$$

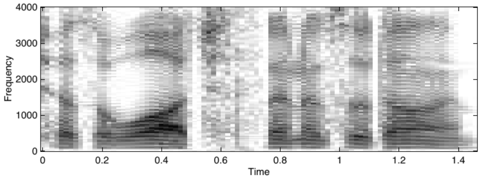
# Improving OLA

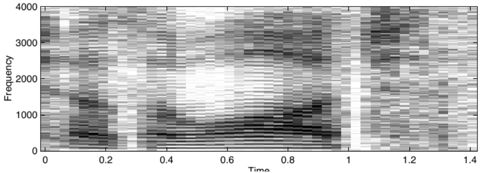
- Phase interactions during overlap of frames can be heard
- Mitigate them by aligning frames prior overlap
-

$$y^m[mL+n] = y^{m-1}[mL+n] + w[n]x[\lceil \frac{m}{r} \rceil L + n + K_m]$$

- find by maximizing cross-correlation

# Time / Frequency approaches




- Why not use the spectrogram ?
- STFT magnitude is not enough...
- We need to recover the phase

# Griffin & Lim algorithm

The complex values of the spectrogram $S$ can be decomposed into polar coordinates, the magnitude spectrogram $|S|$ and the phase spectrogram $\angle S$ under the following relation:

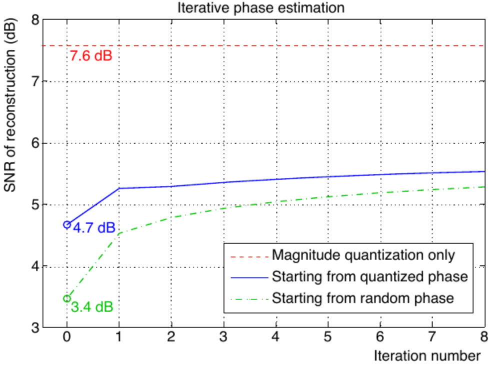
$$S(j,k) = |S(j,k)|e^{i\angle S(j,k)} \tag{1}$$

Assuming an unknown phase set to random values $\angle S_0 = rand$, the algorithm can be iterated until convergence:

1. compute the time domain signal $s_{m-1}$ as the ISTFT of $S_{m-1}$
2. compute the frequency domain representation $S' = \mathrm{STFT}(s_{m-1})$ while enforcing the magnitude spectrogram : $S_m = |S|e^{i\angle S'_m}$

D. Griffin and J. Lim Signal estimation from modified short-time Fourier transform IEEE Tr. Acous,, Speech, and Sig. Proc., 1984

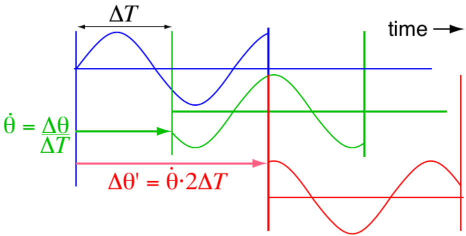
# Griffin & Lim performance




- slow convergence
- but considerable improvement
- need significant overlap between frames

# Phase interpolation

- Principle of the phase vocoder
- Is to interpolate the phase
- Assuming perfect periodicity of the spectral content



# Instantaneous frequency

- Assuming knowledge of the instantaneous frequency
- $$\dot{\Phi}(k,t) = \frac{d}{dt}\angle S(k,t)$$
- Approximate using the frequency of the time / frequency bin:
$$\dot{\Phi}(k,t) \approx k\frac{F_s}{N}$$