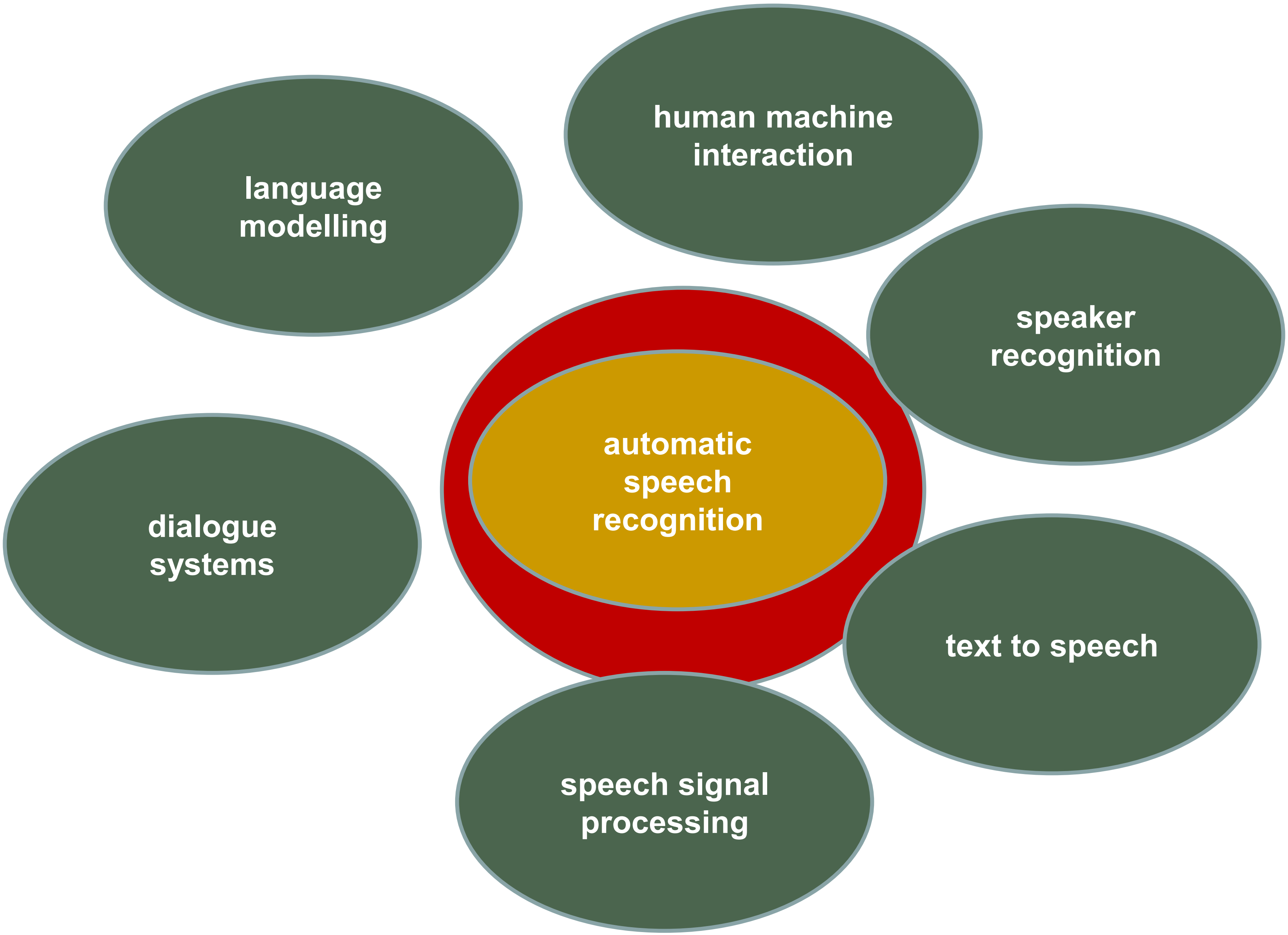
Feature extraction
(audio → vector sequences)
2025-6

Radboud University, Nijmegen

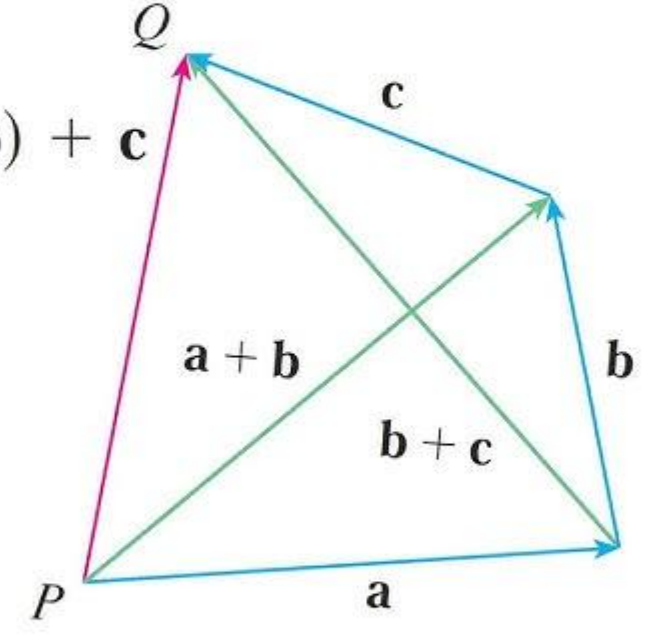
Louis ten Bosch


human machine
interaction
language
modelling
speaker
recognition
automatic
speech
recognition
dialogue
systems
text to speech
speech signal
processing

Big picture: to vectors

Properties of Vectors

1. $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$
2. $\mathbf{a} + (\mathbf{b} + \mathbf{c}) = (\mathbf{a} + \mathbf{b}) + \mathbf{c}$
3. $\mathbf{a} + \mathbf{0} = \mathbf{a}$
4. $\mathbf{a} + (-\mathbf{a}) = \mathbf{0}$
5. $c(\mathbf{a} + \mathbf{b}) = c\mathbf{a} + c\mathbf{b}$
6. $(c + d)\mathbf{a} = c\mathbf{a} + d\mathbf{a}$
7. $(cd)\mathbf{a} = c(d\mathbf{a})$
8. $1\mathbf{a} = \mathbf{a}$



Big picture

Vectorization is a very important step in current approaches (in chatGPT, in end-to-end ASR, in reasoning models, in NLP, in chemistry, ...)

1970: audio → feature vectors (MFCC)
2013: words → vec: word2vec (context independent)
2015 and later: word dependent word embeddings (bank ≠ bank)
2017: attention mechanism → (very) long contexts
2020: wav2vec2.0 (mapping audio to probability vectors on tokens in a dictionary)
2023: whisper (encoder + decoder architecture)

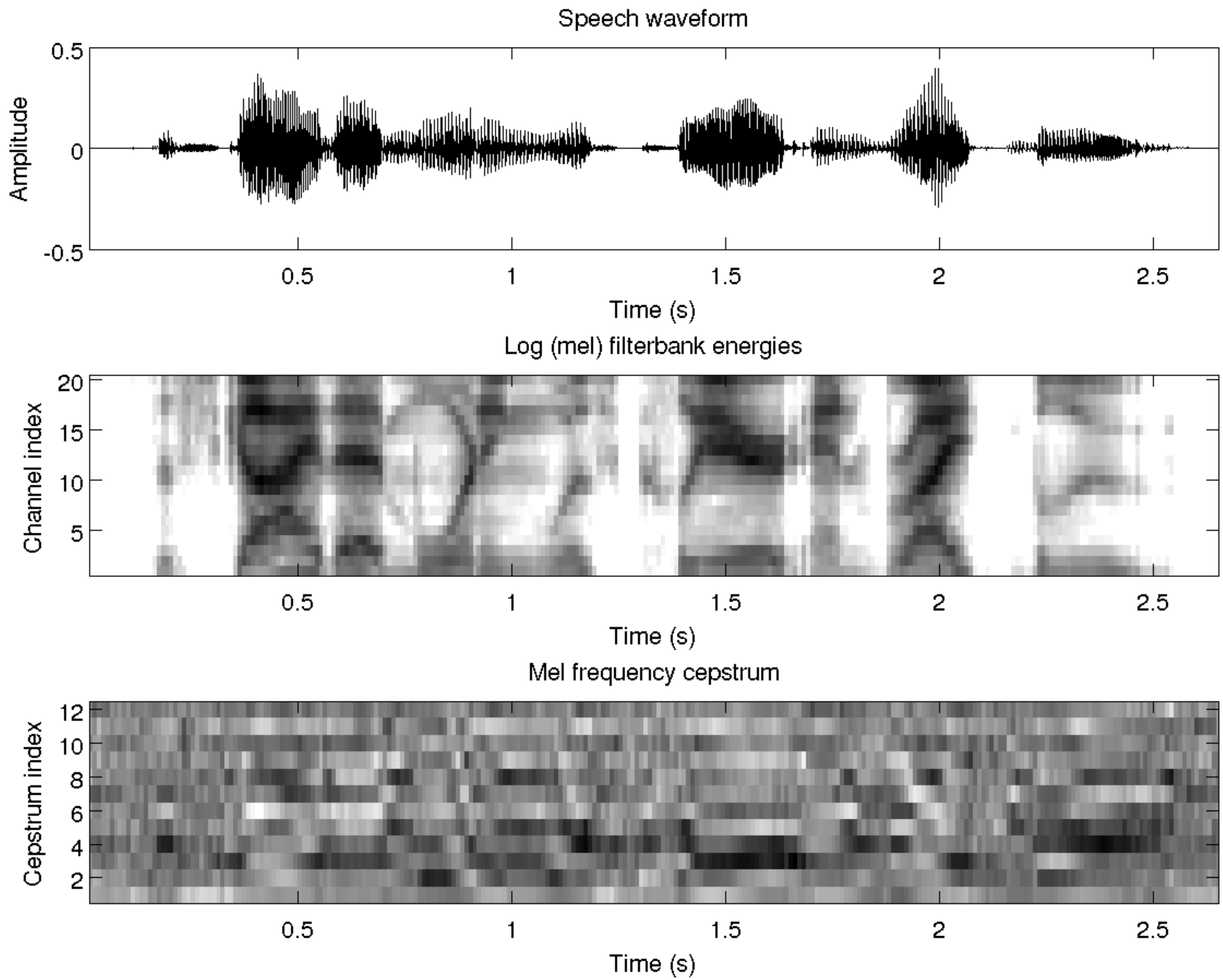
Vectorization → ideal for deep learning and neural networks

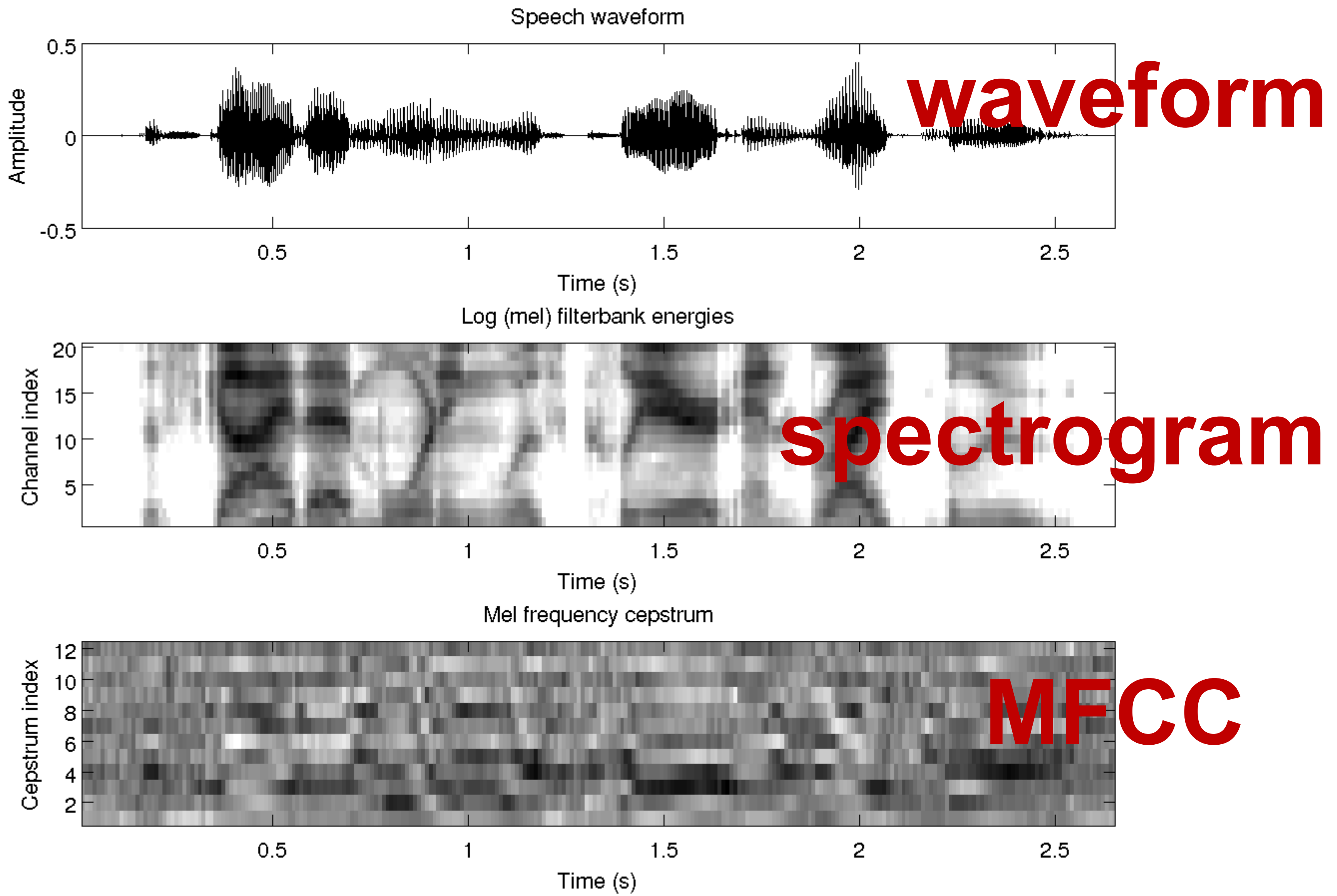
From audio to features

features are input for all
downstream modules

+

what's not in the features cannot
be classified later


Speech waveform
Amplitude
0.5
0
-0.5
0.5
1
1.5
2
2.5
Time (s)
Log (mel) filterbank energies
Channel index
20
15
10
5
Time (s)
Mel frequency cepstrum
Cepstrum index
12
10
8
6
4
2
Time (s)


Speech waveform
Amplitude
0.5
0
-0.5
0.5
1
1.5
2
2.5
Time (s)
waveform
Log (mel) filterbank energies
Channel index
20
15
10
5
Time (s)
spectrogram
Mel frequency cepstrum
Cepstrum index
12
10
8
6
4
2
Time (s)
MFCC

MFCC vectors

MFCC = Mel Frequency Cepstral Coefficients

Very many websites show information about MFCCs, often with useful Python function calls

- https://www.kaggle.com/ilyamich/mfcc-implementation-and-tutorial
- https://pypi.org/project/python_speech_features/
- Librosa python library
  https://librosa.org/doc/latest/index.html

From audio to MFCC

- audio (analog signal) → digital signal
  - AD conversion
- digital signal → MFCC feature vectors
  - in 5 steps

From audio to MFCC

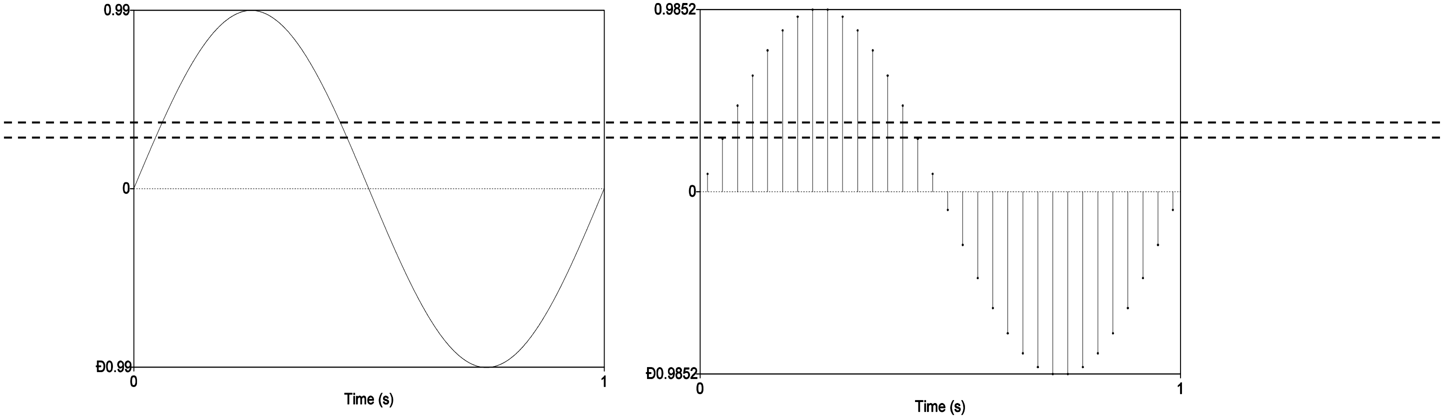
From audio to MFCCs:

0. AD

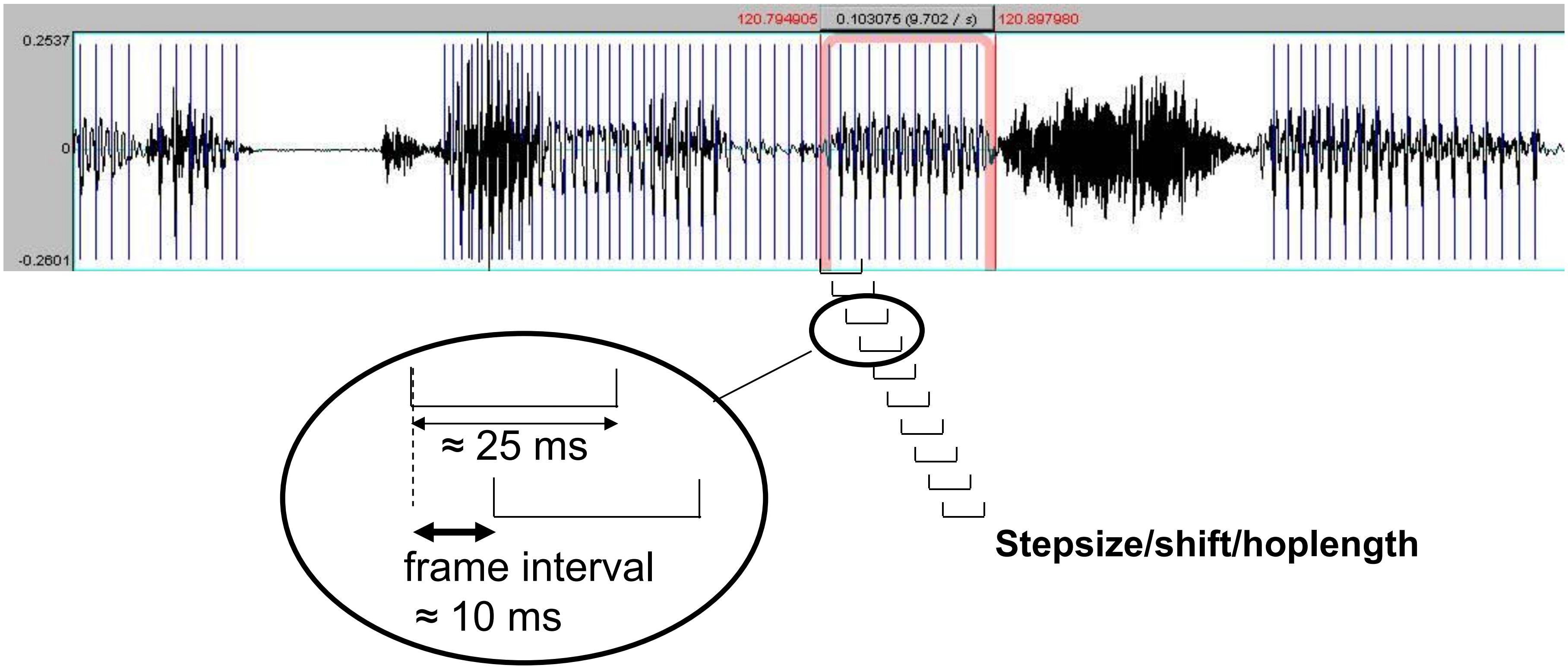
1. Segmentation
2. Smoothing
3. *Fast Fourier Transform (FFT):* Conversion from the time domain to the frequency domain
4. Apply perceptual weighting based on human auditory processing.
5. Decorrelation

0 analog-digital (AD) conversion

- Discretisation in time
  - sampling frequency or sampling rate (samples/sec, Hz) determines the highest frequency that can be represented. **Nyquist**. (10 kHz-44 kHz)
- Discretisation in amplitude
  - Number of possible amplitude values is determined by bytes/sample, e.g.
    - 8 bits (1 byte): 2^8 (256) possible values
    - 16 bits (2 bytes): 2^{16} (65536) possible value



1 segmentation



25ms: analysis window length

10ms: frame shift/stepsize/hoplength

If sample freq = 16kHz, 25ms corresponds to 0.025*16000 = 400 samples.

1 what matters in segmentation?

what is being said ←→ shape of the vocal tract

shape of the vocal tract ←→ the energy envelope of the spectrum

What is a reasonable analysis duration ?

The average duration of a speech sound is 70ms.

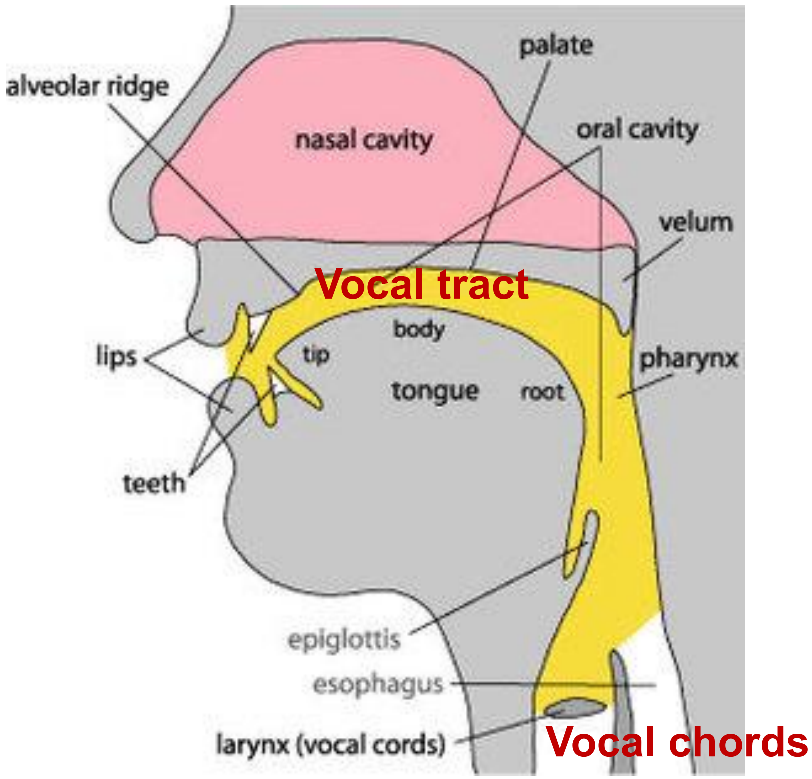
A defendable analysis frame duration is 25 ms.

What about the shift?

To accurately describe the changes in vocal tract shape over time, the number of analyses per second must be at least twice as high as the highest frequency with which the vocal tract changes (Nyquist criterion).

100 times/second (i.e. every 10 ms) is enough

1 segmentation: vocal tract



Articulation is relatively slow

About 12-14 speech sounds per second, i.e. 70 ms. per phone, on average

Articulations move synchronously/in parallel → assimilation of properties of neighboring sounds

2 smoothing/windowing

Hard boundaries give audible artefacts. These artefacts can be avoided by proper windowing: taper off the beginning and end of the signal.

For a well-chosen window, the spectrum is nearly identical to a signal of which the core part is repeated indefinitely.

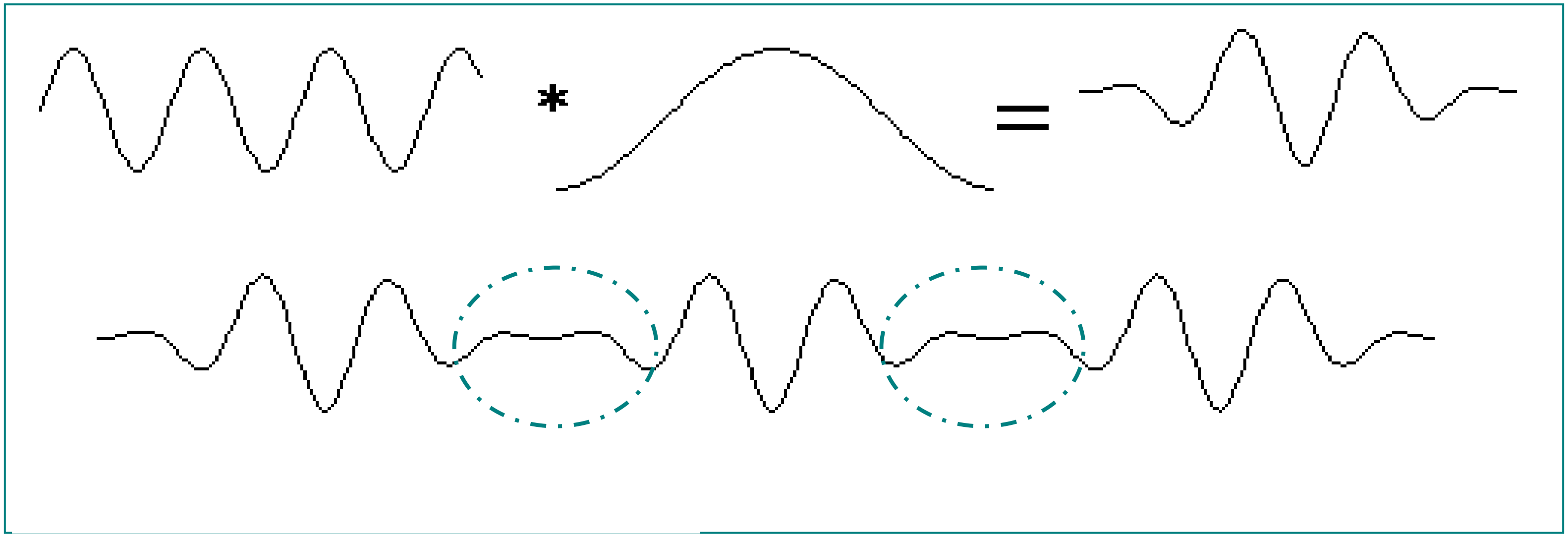
Often used windows are Hamming and Hanning windows. See e.g. https://www.youtube.com/watch?v=YsqGQzJ_2V0

2 smoothing/windowing

segmented waveform

window

windowed waveform



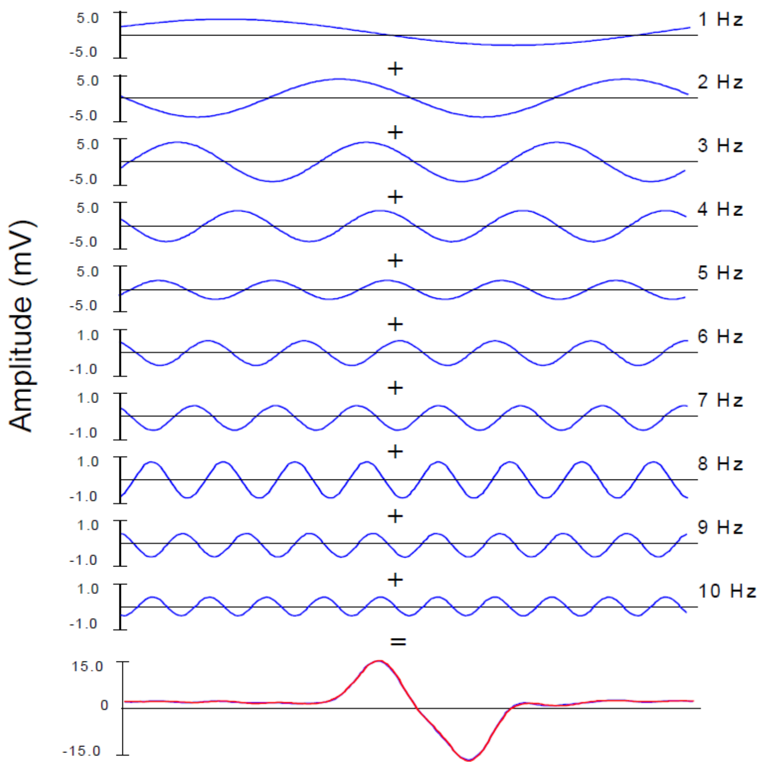
Source figures: http://www.bores.com/courses/intro/freq/3_window.htm

3 FFT

Fast Fourier transform (FFT): maps time domain to frequency domain

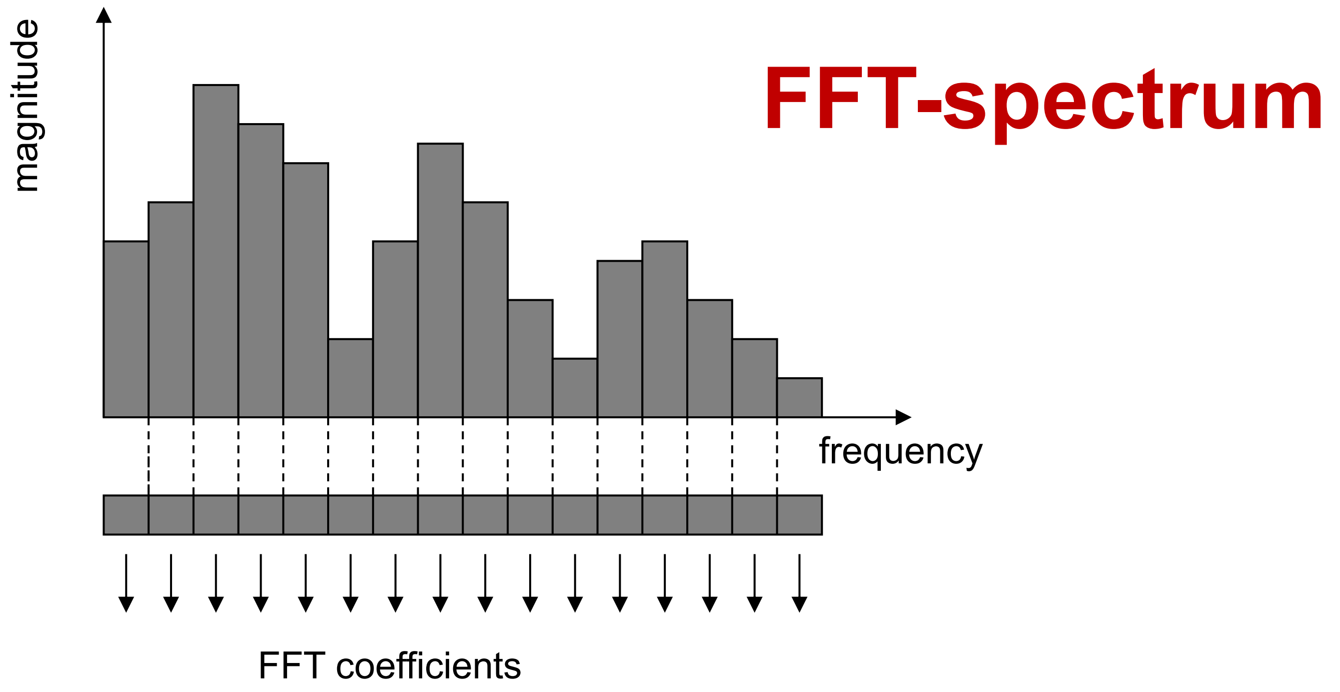
Jean-Baptiste Fourier: Every waveform is the sum of sine waves with a certain magnitude and phase

The signal in red is decomposed in terms of a weighted sum of sine waves with frequencies 1, 2, 3, 4 …



3 FFT

Output:



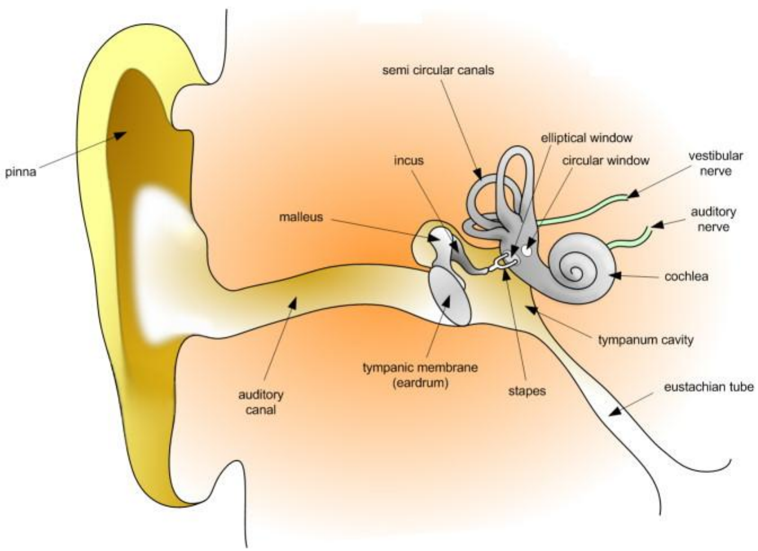
3 FFT

The resulting FFT coefficients are written in one vector

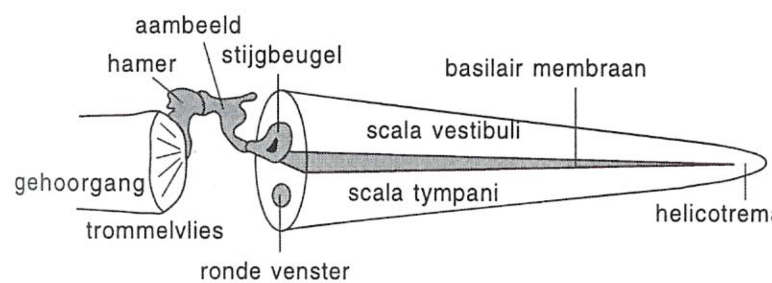
- the more coefficients, the more accurate the description
- but: higher-order coefficients may be noisy

4 perceptual weighting

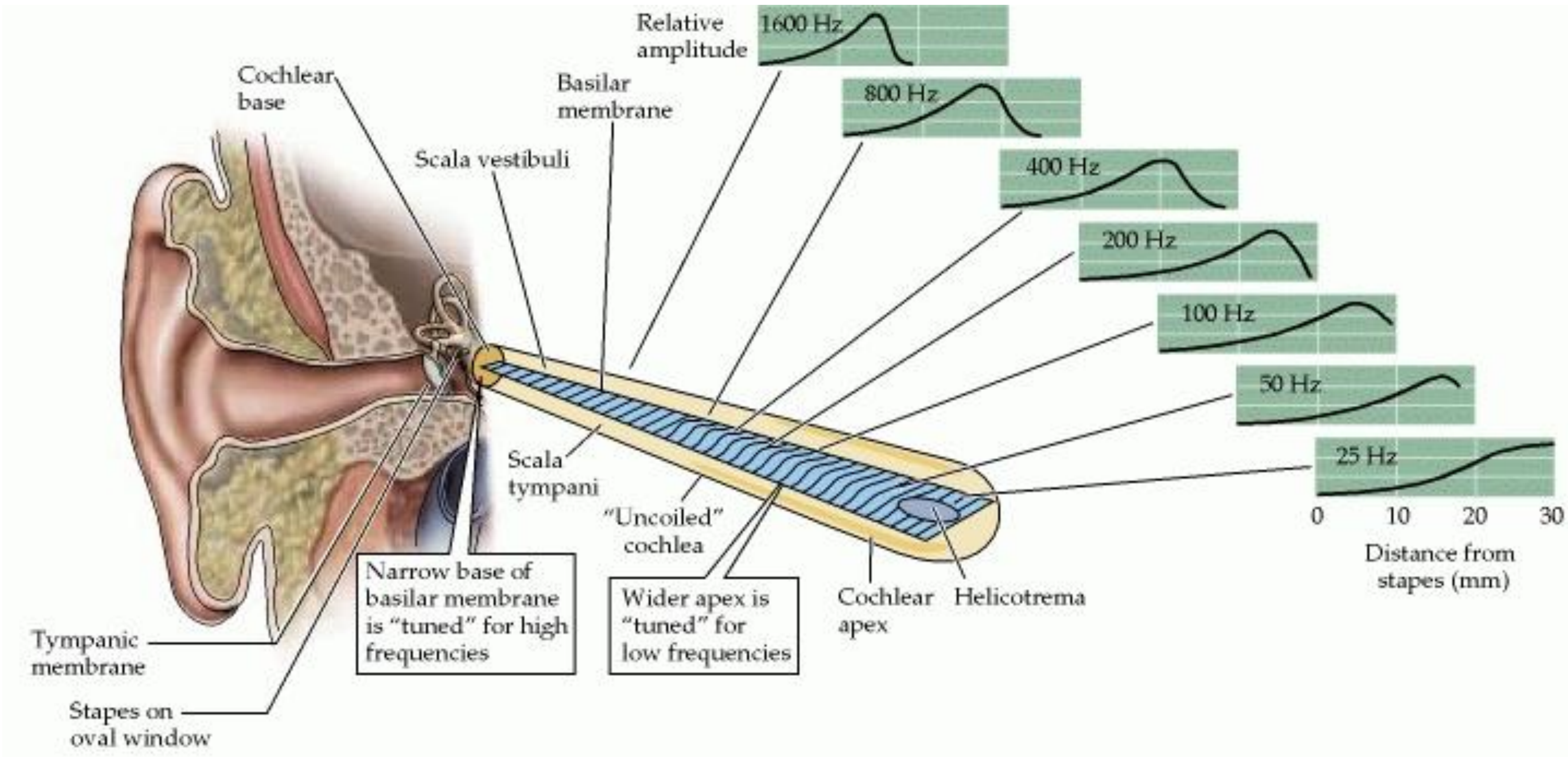
Convert FFT coefficients using perceptual properties of the human auditory system.



malleus, incus, stapes



Cochlea in normal (left, wikipedia) en unrolled form



Weber's law https://www.youtube.com/watch?v=hHG8io5qIU8

Physics	**Perception**
Energy as function of frequency	**log(E) as function of log(f)**

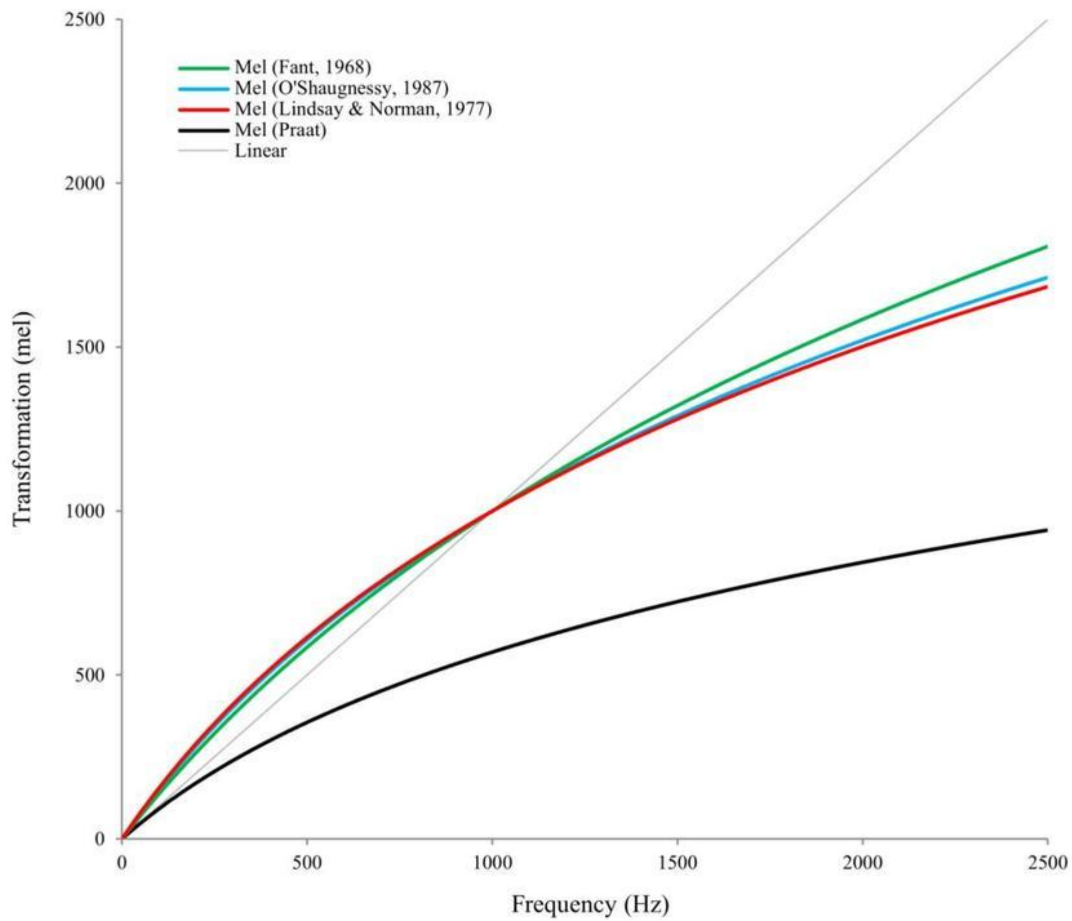
4 perceptual weighting

- The human ear is not sensitive to frequency along a linear scale
- Psycho-acoustical frequency scales often used to approximate the human non-linear sensitivity
- Examples: the *Mel scale, Bark scale*

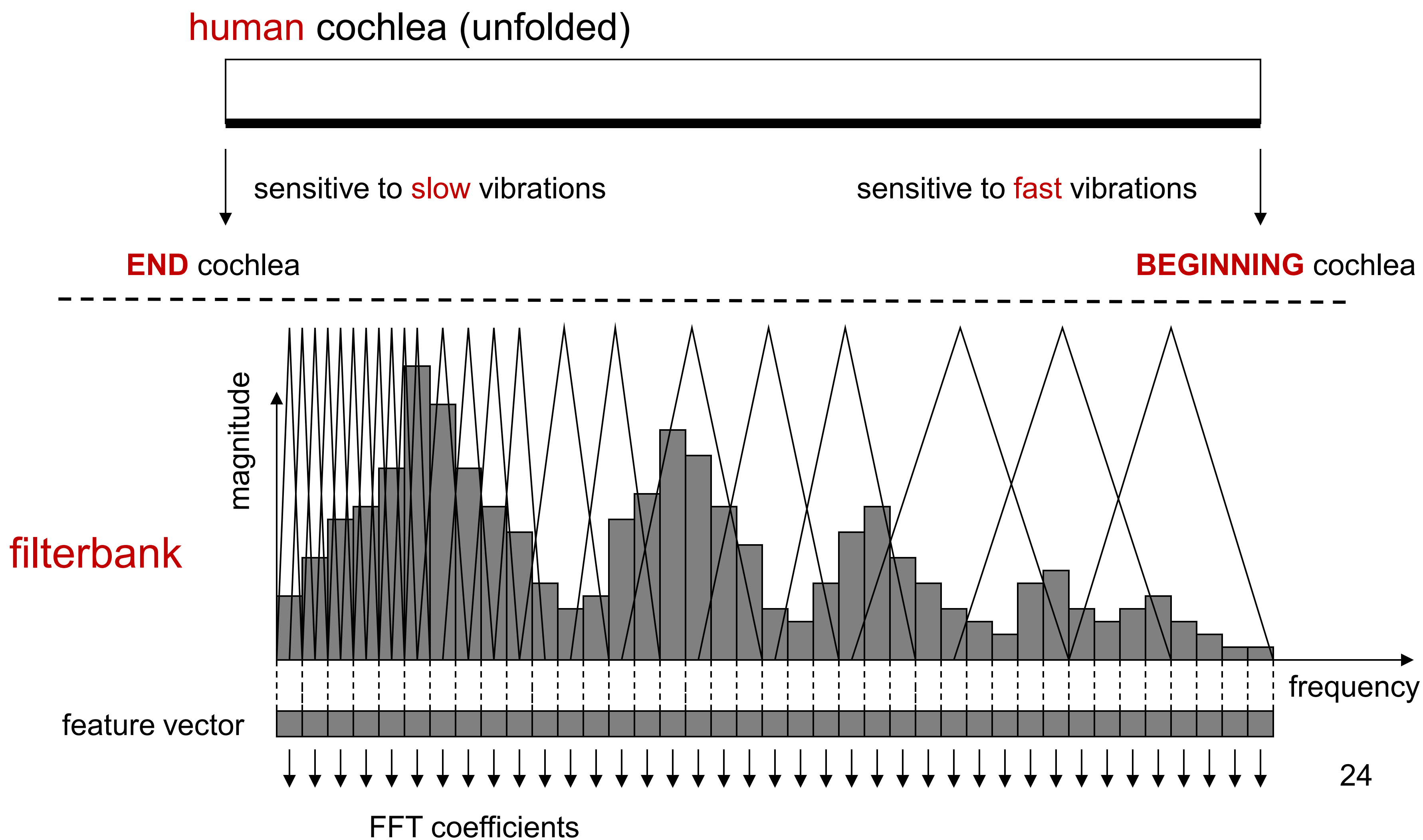
4 frequency to mel: f→ mel(f)

$$\text{mel}(f) = 1125 \log(1+f/700)$$

There are several other transformations, all log-like.



4 mel-filterbank



4 log() all energy values in each filter

- E → log(E)

$$\Delta Percept = \frac{\Delta PhysicalQuantity}{PhysicalQuantity}$$

- **Weber's law**
- **https://www.youtube.com/watch?v=hHG8io5qIU8**
  - Energy → loudness
  - Fundamental frequency → pitch (piano)
  - Perception of physical phenomena
  - Estimation of physical quantities
  - Duration, length, pressure, …

5 decorrelation

The amount of energy in neighbouring filters is strongly correlated. In order to reduce this correlation, a Discrete Cosine Transform (DCT) is performed

We obtain the **Mel Frequency Cepstral Coefficients (MFCCs).**

These MFCCs are approximately statistically independent. The first 12 coefficients $c_1..c_{12}$ suffice to describe the relevant details of the spectrum (for that analysis window).

See e.g.
https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html for comments on the DCT step

Audio to MFCC: summary

Summary

		output	typical size
0.	A/D conversion	digital signal	16000/sec
1.	segmentaton	analysis stretch	400 samples/10ms
2.	smoothing	windowed signal	400 samples/10ms
3.	FFT	spectrum (A(f))	400 magnitudes/10ms
4.	filterbank	feature vector	20-40 energies/10ms
5.	decorrelation	feature vector	12 features/10ms

assuming 16kHz, 25 ms analysis frame

Alternative techniques to extract features from the speech signal

Techniques to extract features from speech signal	
Principal Component Analysis (PCA)	Linear map, fast, eigenvector-based, Traditional, eigenvector base method, OK for Gaussian data
Linear Discriminate Analysis (LDA)	Supervised linear map; fast, eigenvector-based Better than PCA for classification
Independent Component Analysis (ICA)	Linear map, iterative non-Gaussian, blind course separation, used for de-mixing non-Gaussian distributed sources
Linear Predictive Coding	Aiming at dim reduction, 10 to 16 coefficients
Cepstral Analysis	Represents shape of spectral envelope in power domain
Filter bank analysis	Uses filters tuned to specific frequencies
Mel-frequency cepstral coeff (MFCCs)	**Fourier Analysis, filter bank, human auditory pathway**
Kernel based feature extraction	Dimensionality reduction, reduces redundancy in features
Wavelet	It replaces the fixed bandwidth of Fourier transform with one proportional to frequency

Newer features

- MFCC (Mel-Frequency Cepstral Coefficients)
- TECC (Teager-Energy Cepstral Coefficients)
- TEMFCC (Teager-based Mel-Frequency Cepstral Coefficients)
- Features via Deep Denoising AutoEncoders (DDAE)
  - E.g. for generating whisper-robust cepstral features