

Unit 3: Application Areas

3b: Audition

Audition

- Vision: use eyes/camera and extract perceptual information
- Audition: use ears/microphone and extract perceptual information
- A little experiment...




Audition tasks

- Computational Auditory Scene Analysis (CASA)
 - Separate out different sound signals
- Speech enhancement
 - Similar to CASA, but focused on extracting speech from other signals (noise, other speech)
- Music analysis
 - Analysis of musical styles
 - Music information retrieval

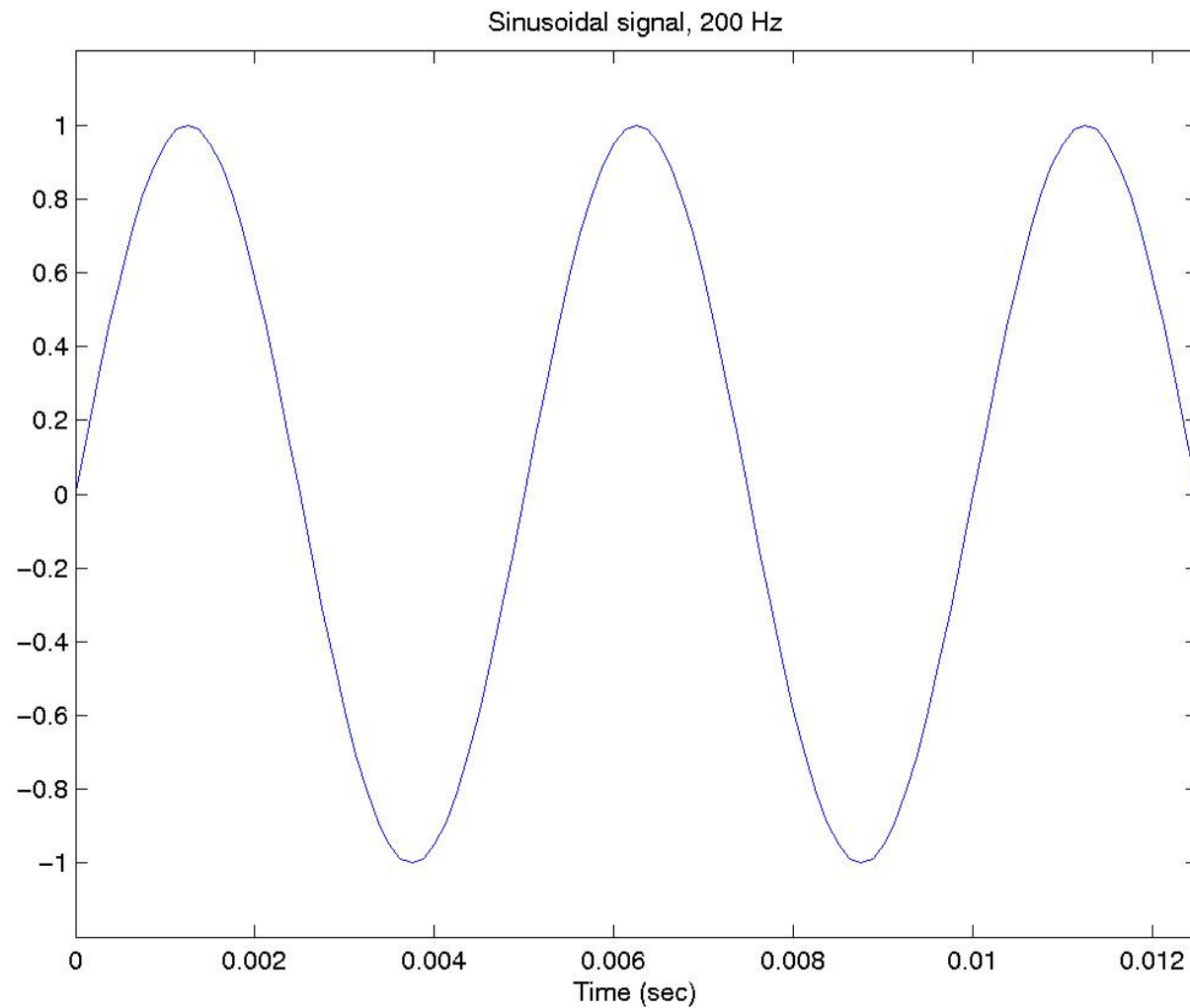
Pitch

- Many of these applications depend on pitch
- Simplest definition:
 - Sounds are vibrations of air molecules at a particular frequency
 - frequency = $1/\text{wavelength}$
 - 1 Hertz (1 Hz) = 1 vibration/sec
 - Often, vibrations occur at multiples of the fundamental frequency
 - Strength of *overtones* gives instruments different sounds, for example
 - Pitch is (to a first approximation) the fundamental frequency

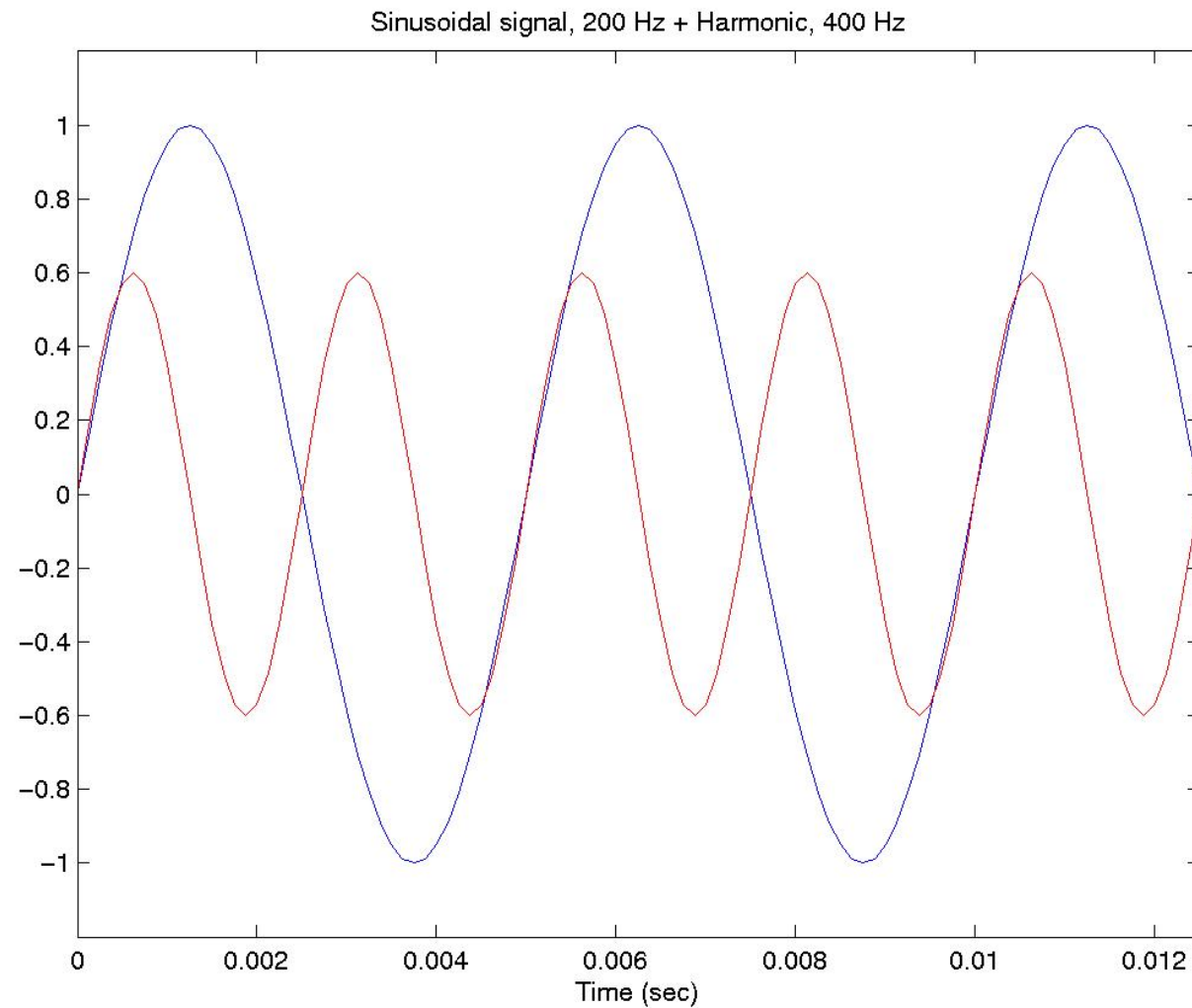
Pitch as Percept

- We don't necessarily need the fundamental frequency to “get” a pitch percept.
- We can fool our auditory system into thinking that a pitch exists by changing energies of different harmonics within a sound, or making it sound like there is a pitch structure when there isn't.
 - We can remove low harmonics and keep the pitch the same 
 - We can change harmonic energies to create never-ending rising or falling sequences 
 - We can sample/filter noise to make it sound like it has a pitch 

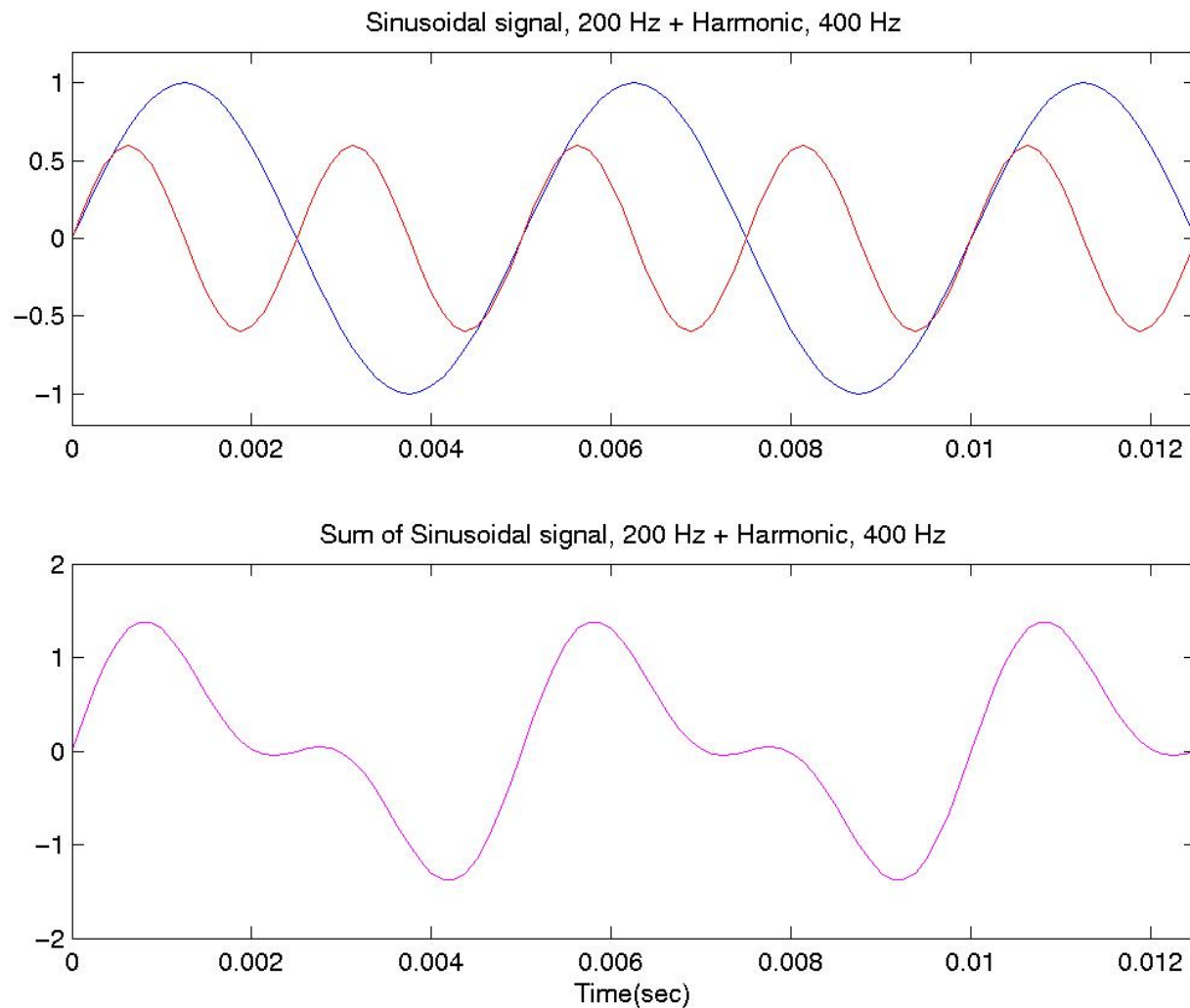
Fundamental frequency @ 200 Hz



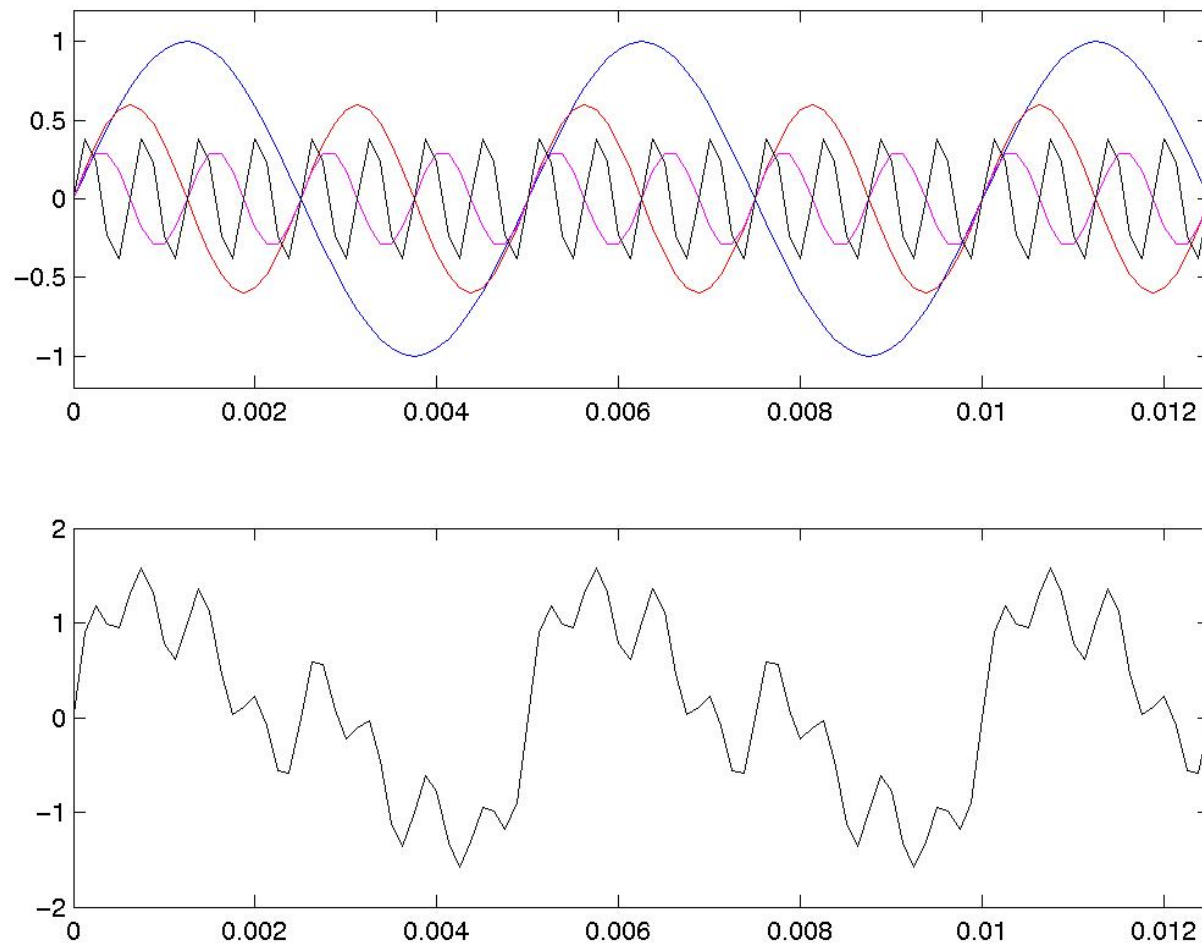
Adding in a harmonic @ 400Hz



Complex signal: Summing the two harmonics



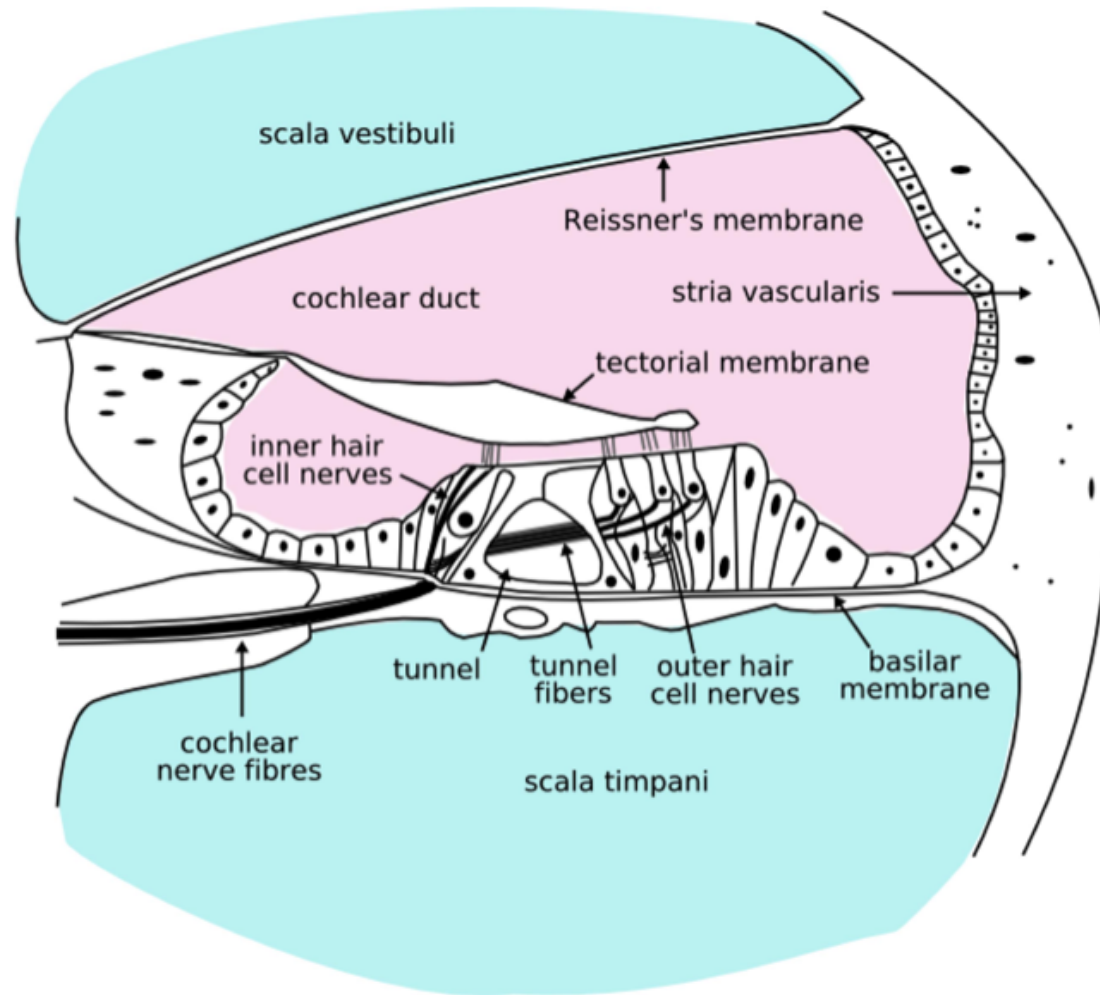
A yet more complex signal: 4 harmonics



How humans do it

- The ear has several parts that allow it to pick up on acoustic frequencies
 - Eardrum: transmits sounds from outer ear to 3 bones in middle ear (ossicles)
 - Ossicles transmit sound vibrations to cochlea
 - Cochlea: a fluid-filled snail-shaped tube
 - Hair cells: act as frequency triggers inside cochlea
 - Cochlea is organized tonotopically (in frequency order): high frequency near ossicles, low frequency further down basilar membrane

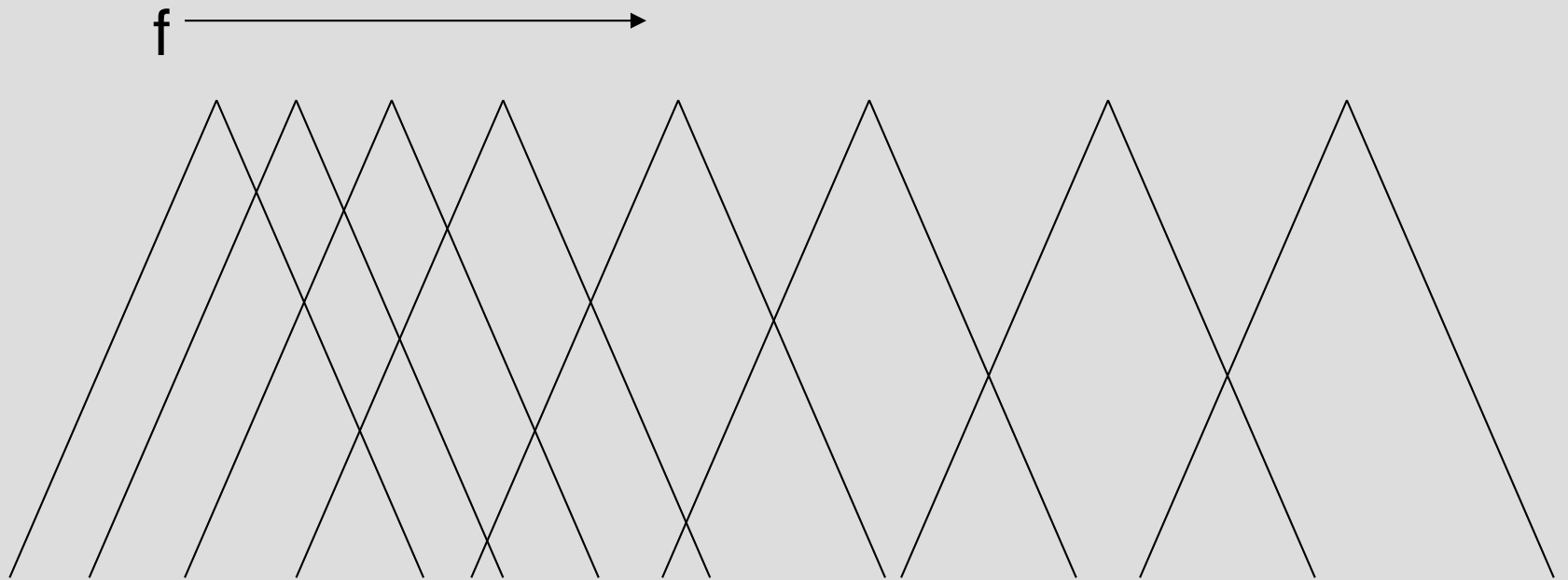
A cross-section of the cochlea



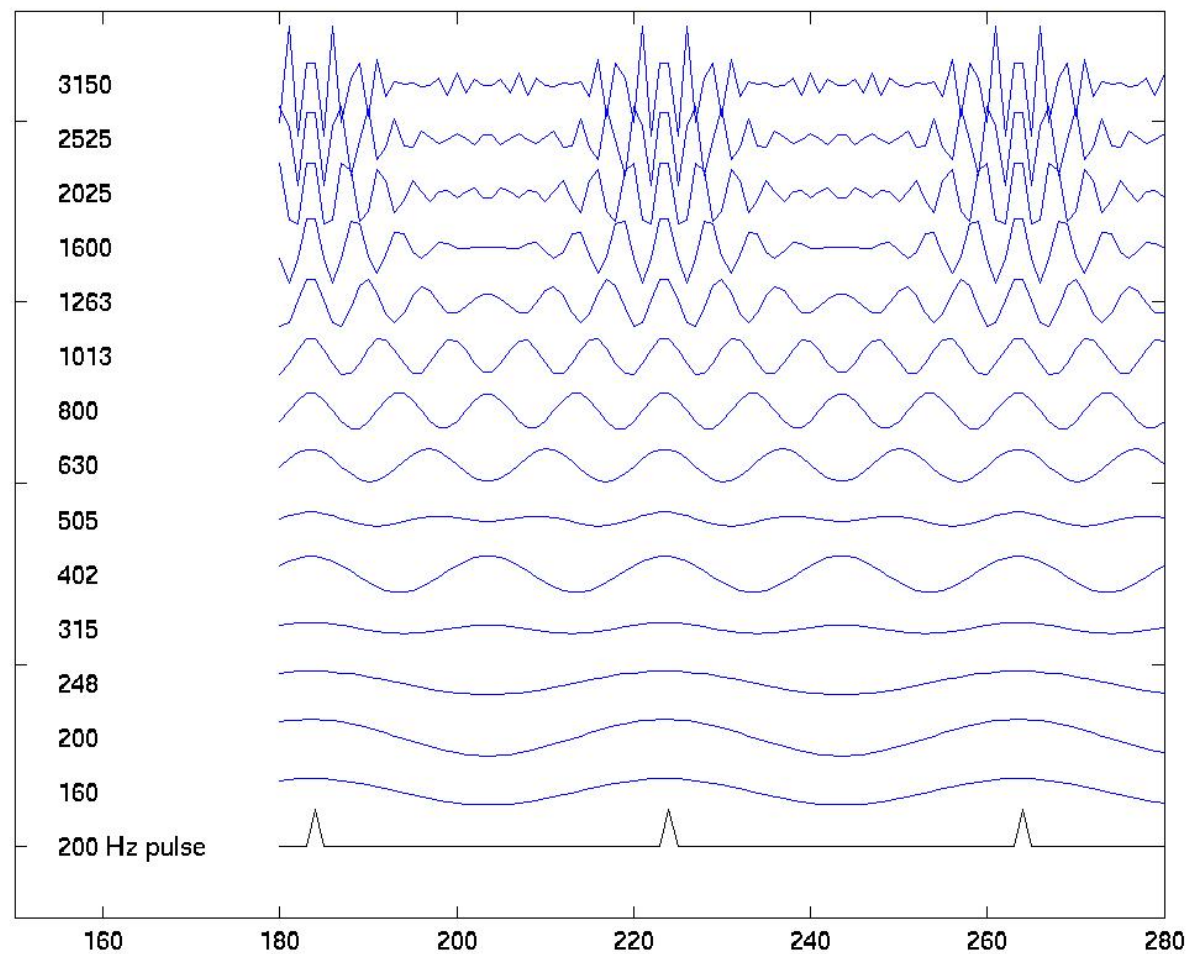
source: Wikipedia

Machine approximations to the cochlea

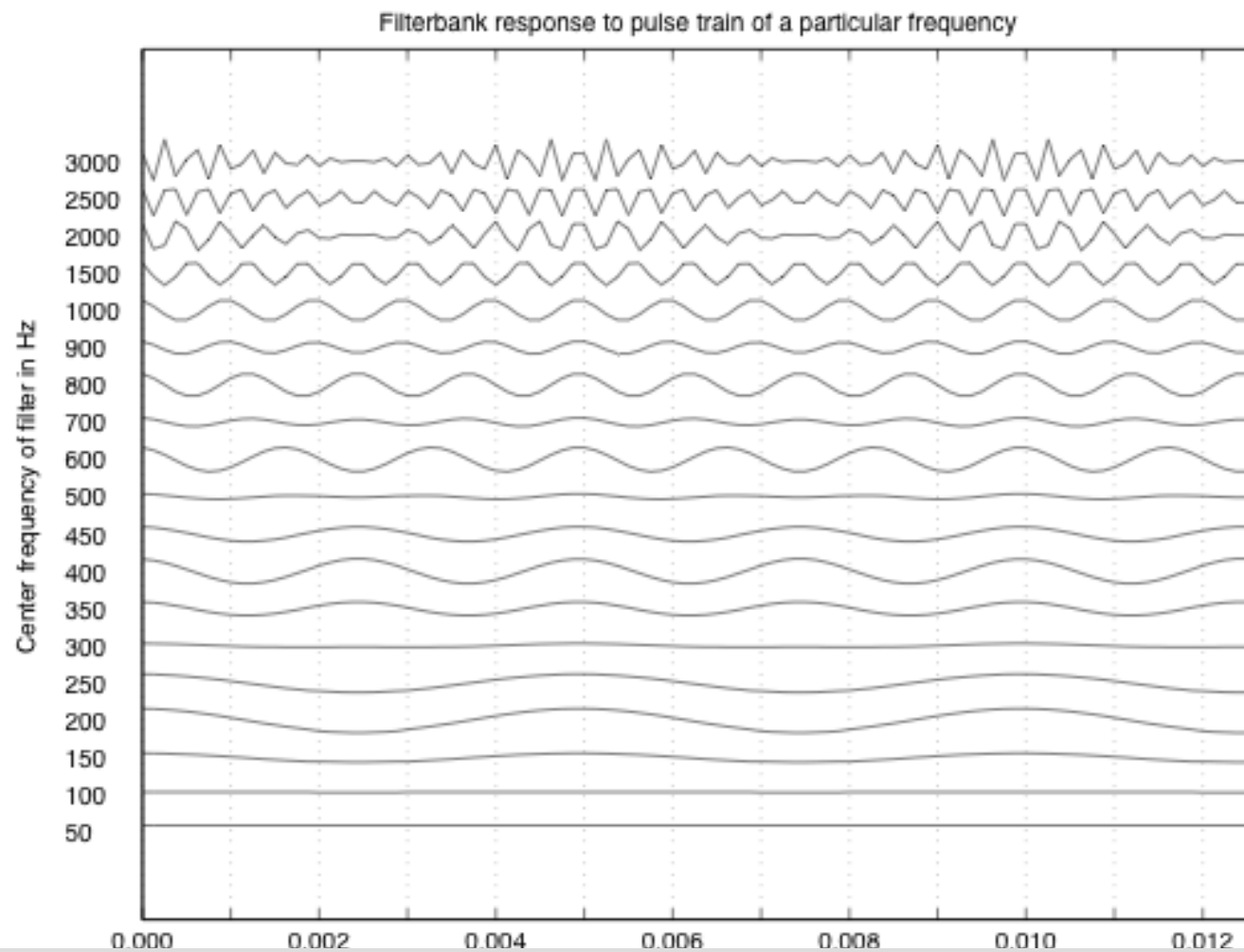
- Filterbank: a bunch of overlapping frequency filters centered at various points along the frequency spectrum



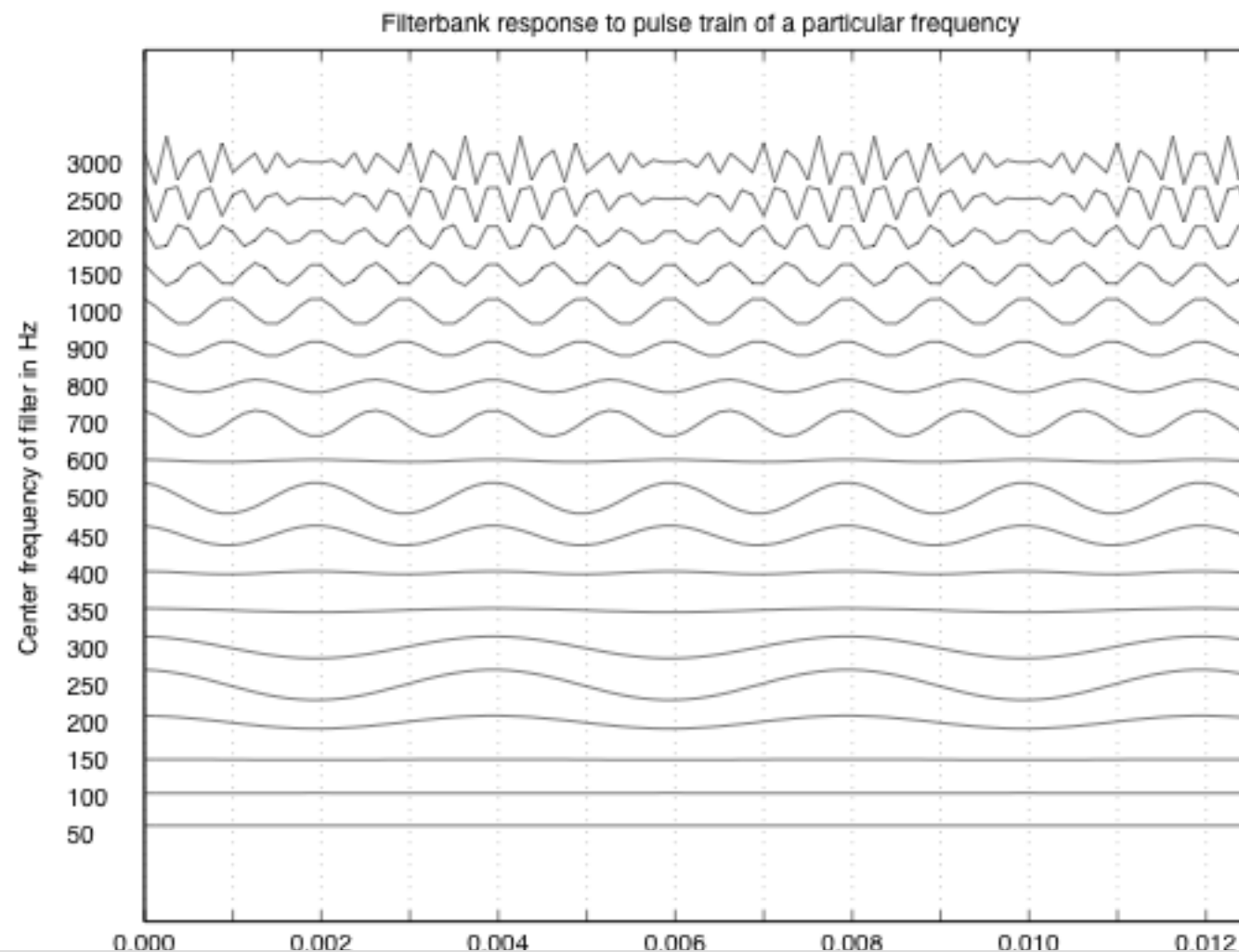
Filterbank response to 200 Hz Pulse



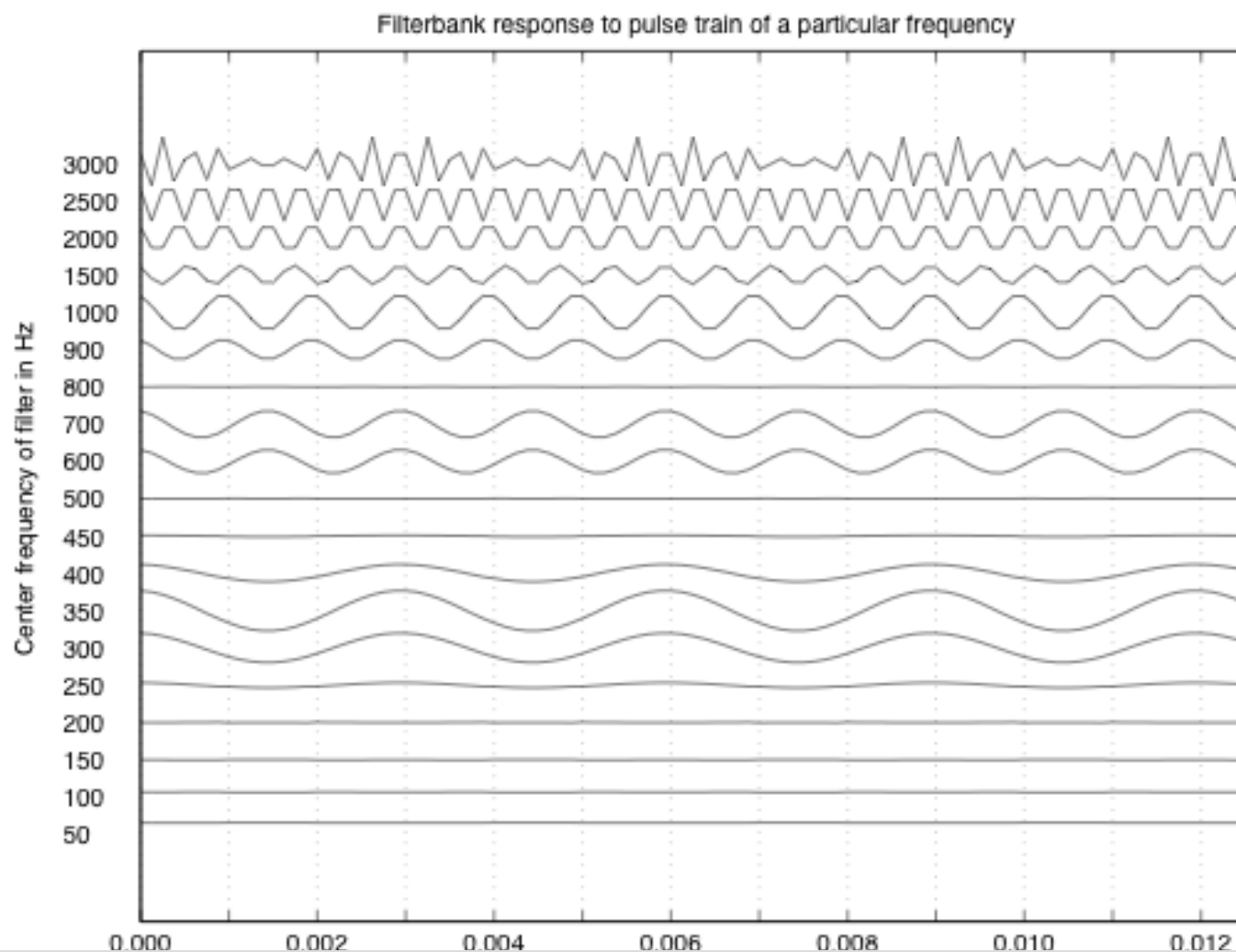
200 Hz, different filterbank



250 Hz, same filterbank



333 Hz, same filterbank



Autocorrelation

- You can (usually) find the fundamental frequency by comparing a signal to shifted versions of itself (Autocorrelation)
- $R(j) = \sum x_n x_{n-j}$



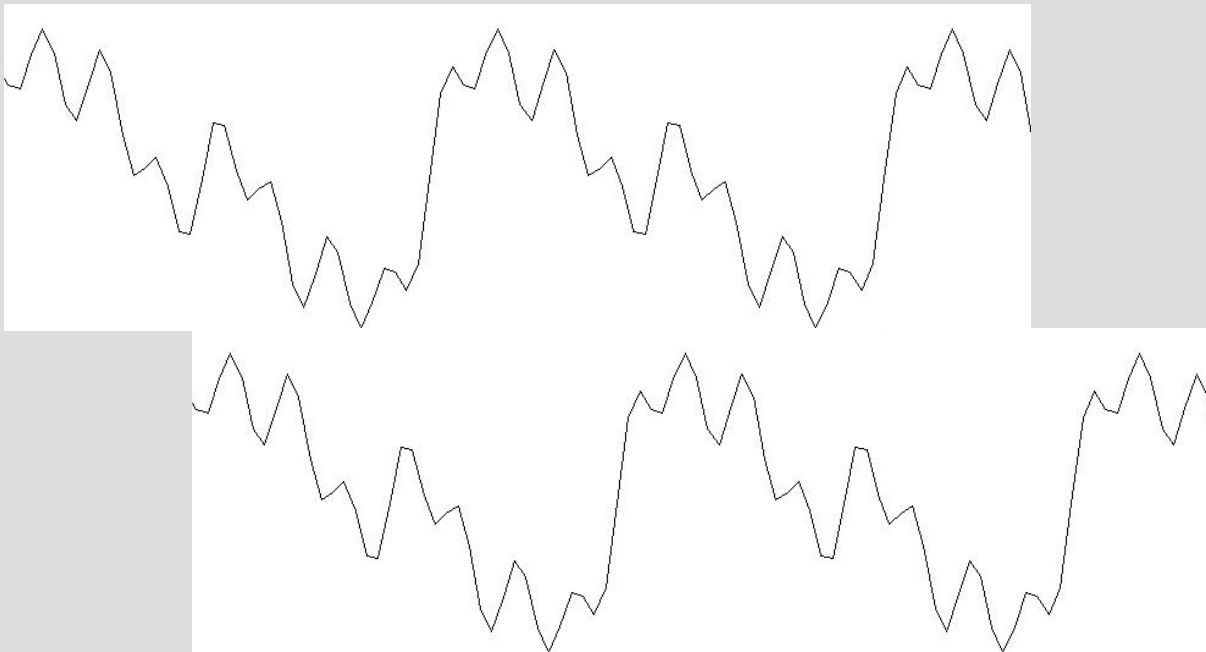
Autocorrelation

- You can (usually) find the fundamental frequency by comparing a signal to shifted versions of itself (Autocorrelation)
- $R(j) = \sum x_n x_{n-j}$



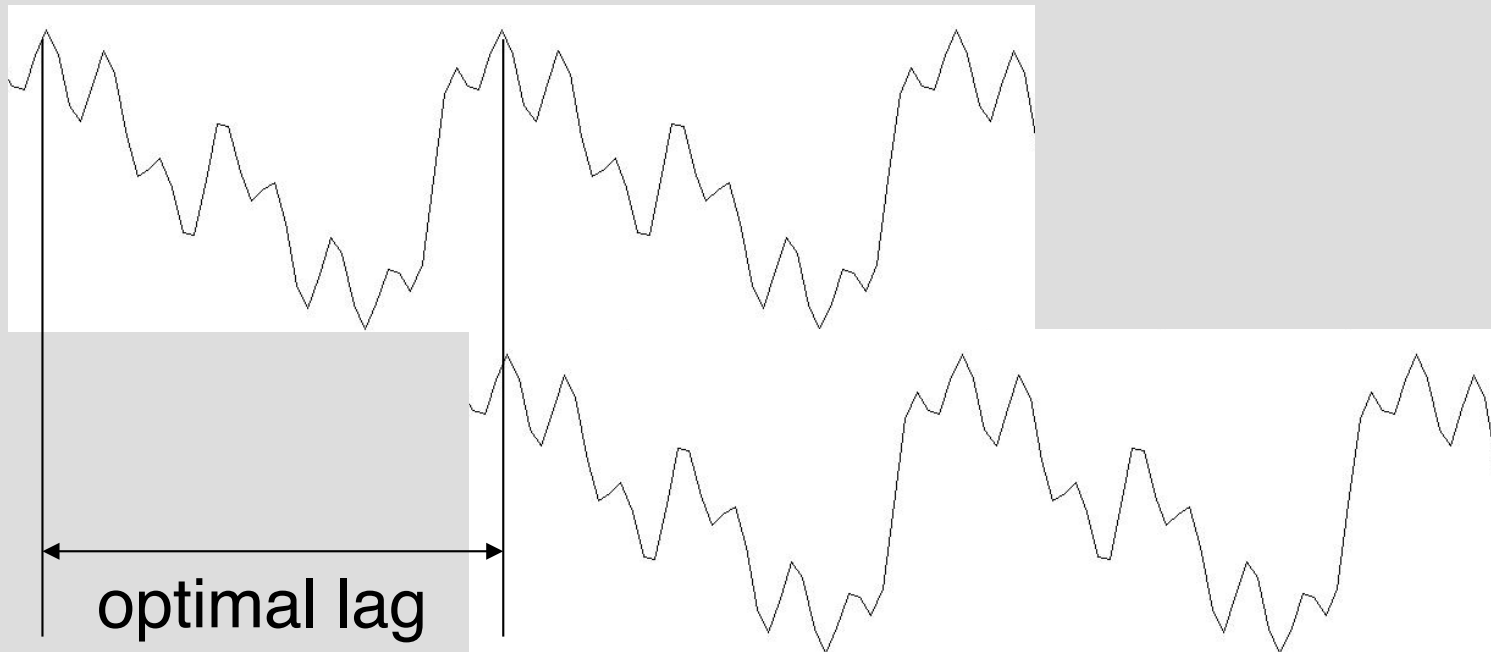
Autocorrelation

- You can (usually) find the fundamental frequency by comparing a signal to shifted versions of itself (Autocorrelation)
- $R(j) = \sum x_n x_{n-j}$



Autocorrelation

- You can (usually) find the fundamental frequency by comparing a signal to shifted versions of itself (Autocorrelation)
- $R(j) = \sum x_n x_{n-j}$



Mid-level Auditory Representations

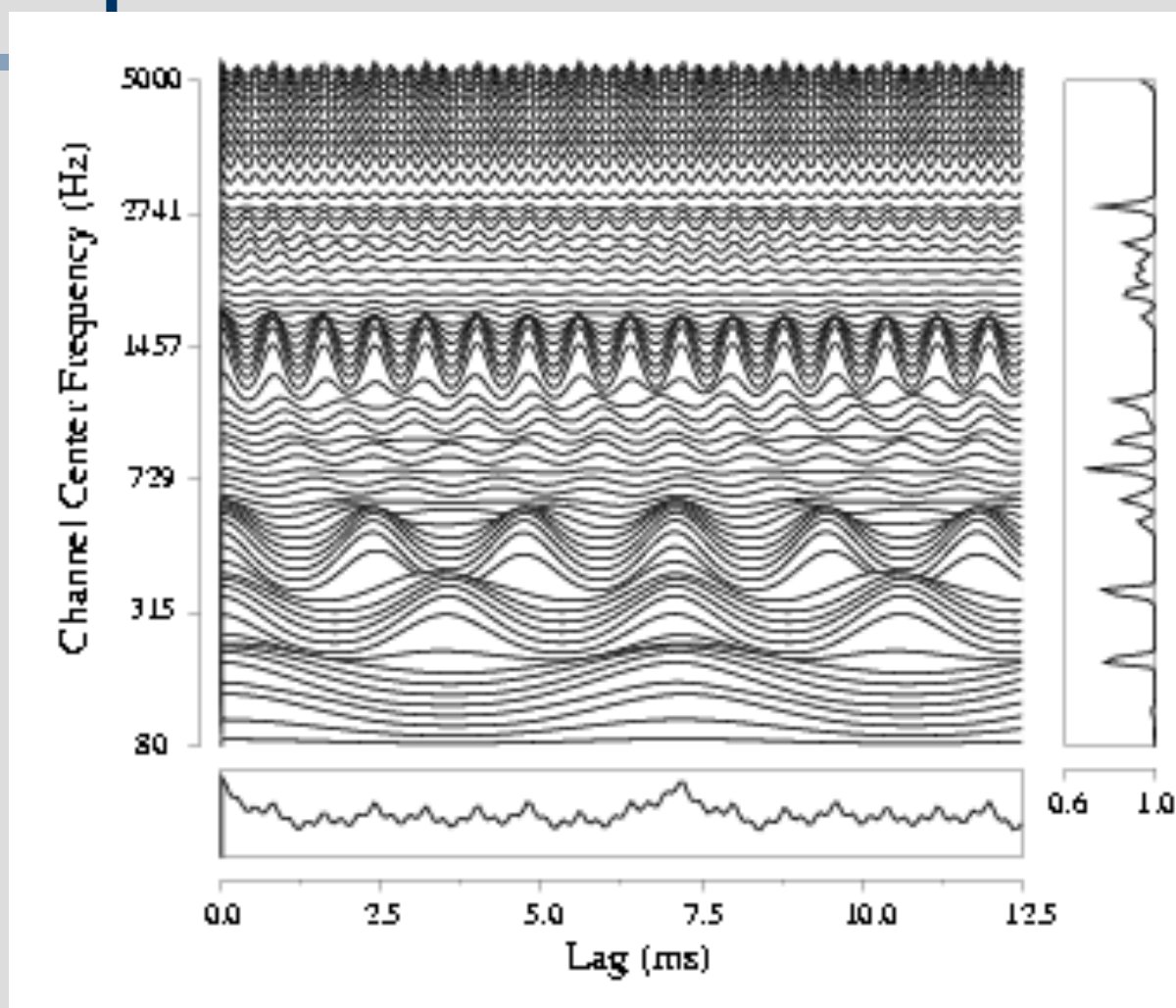
Mid-level representations form the basis for segment formation and subsequent grouping processes

Correlogram extracts periodicity information from frequency analysis

Summary correlogram can be used to identify F0

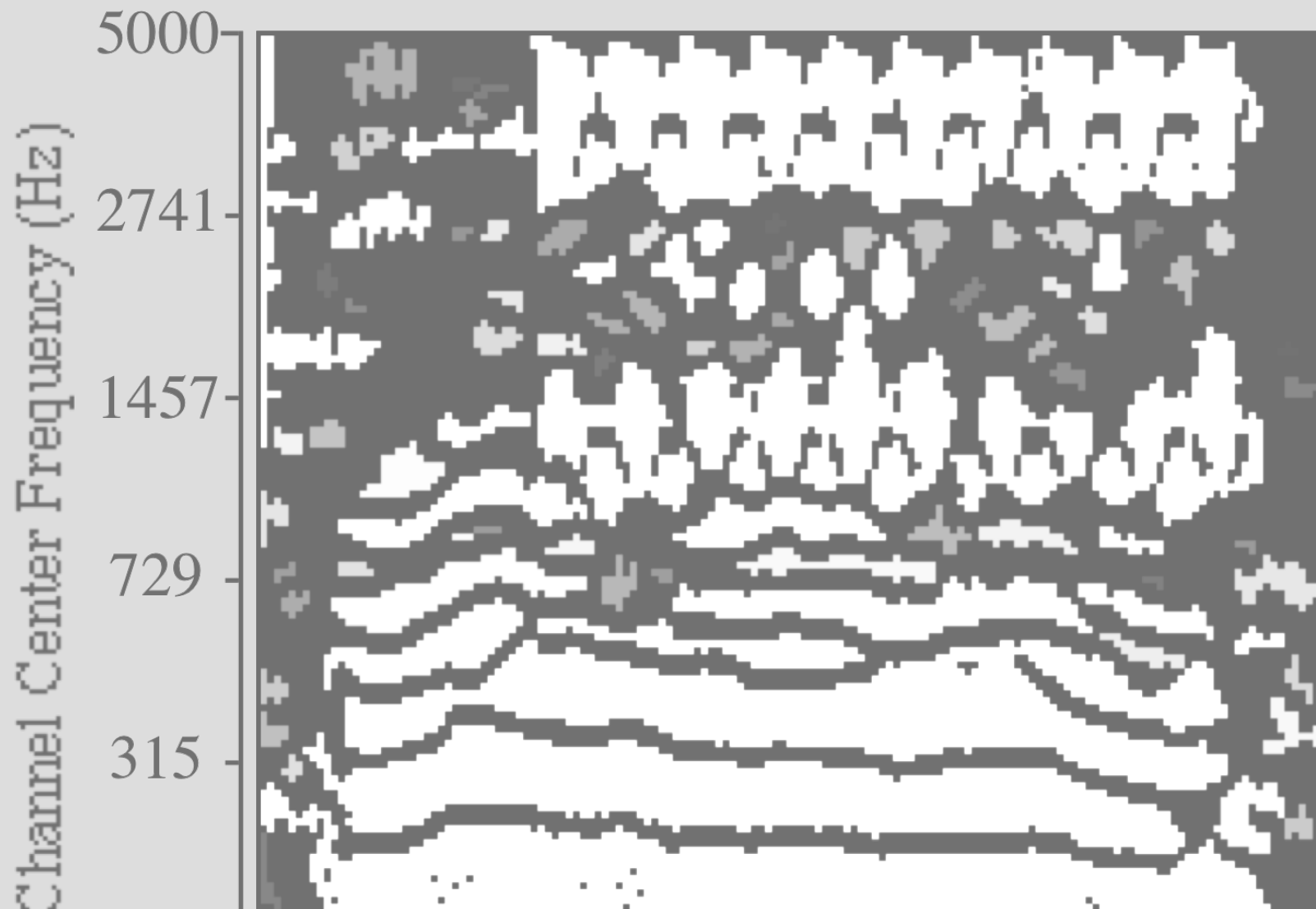
Cross-correlation between adjacent correlogram channels identifies regions that are excited by the same frequency component

Mid-level Representations - Example

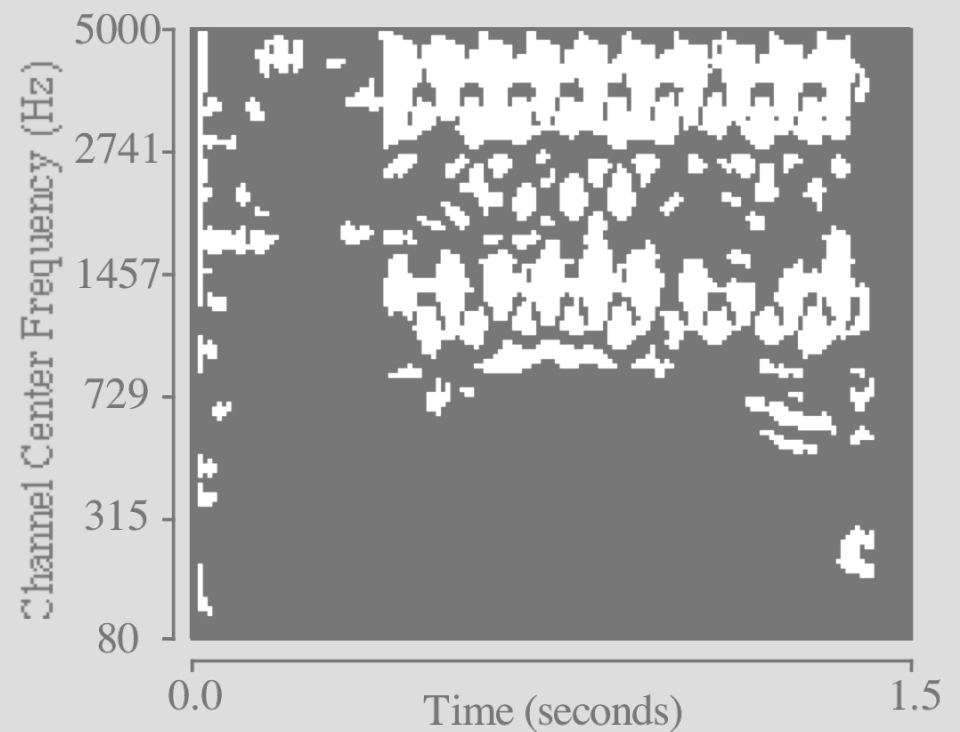
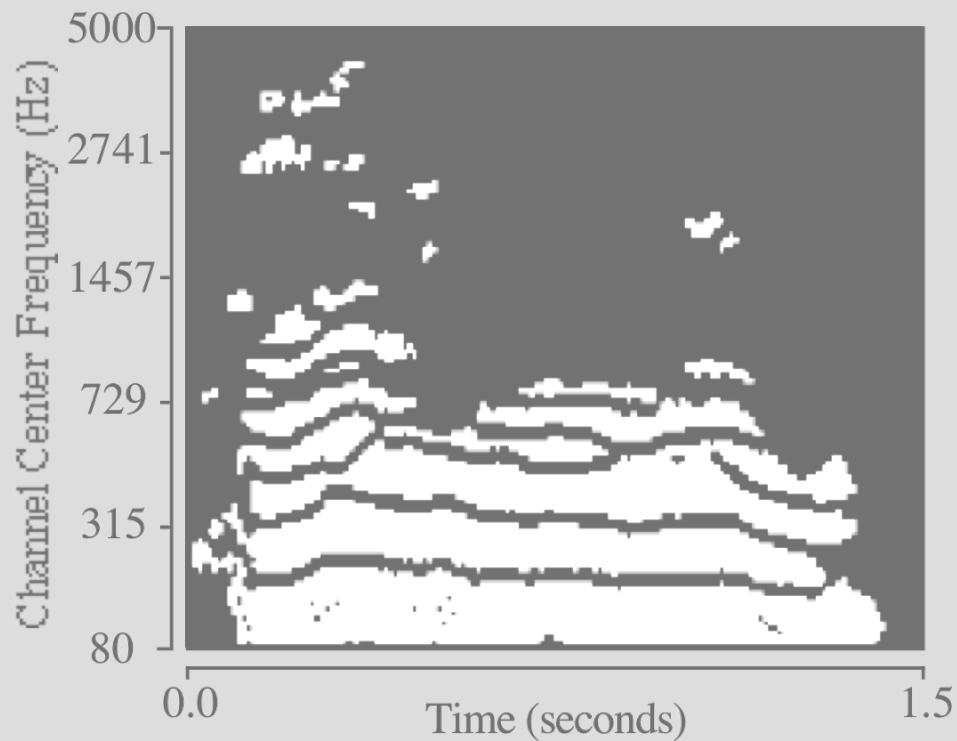


Correlogram and cross-correlation for speech/telephone mixture

Segmentation



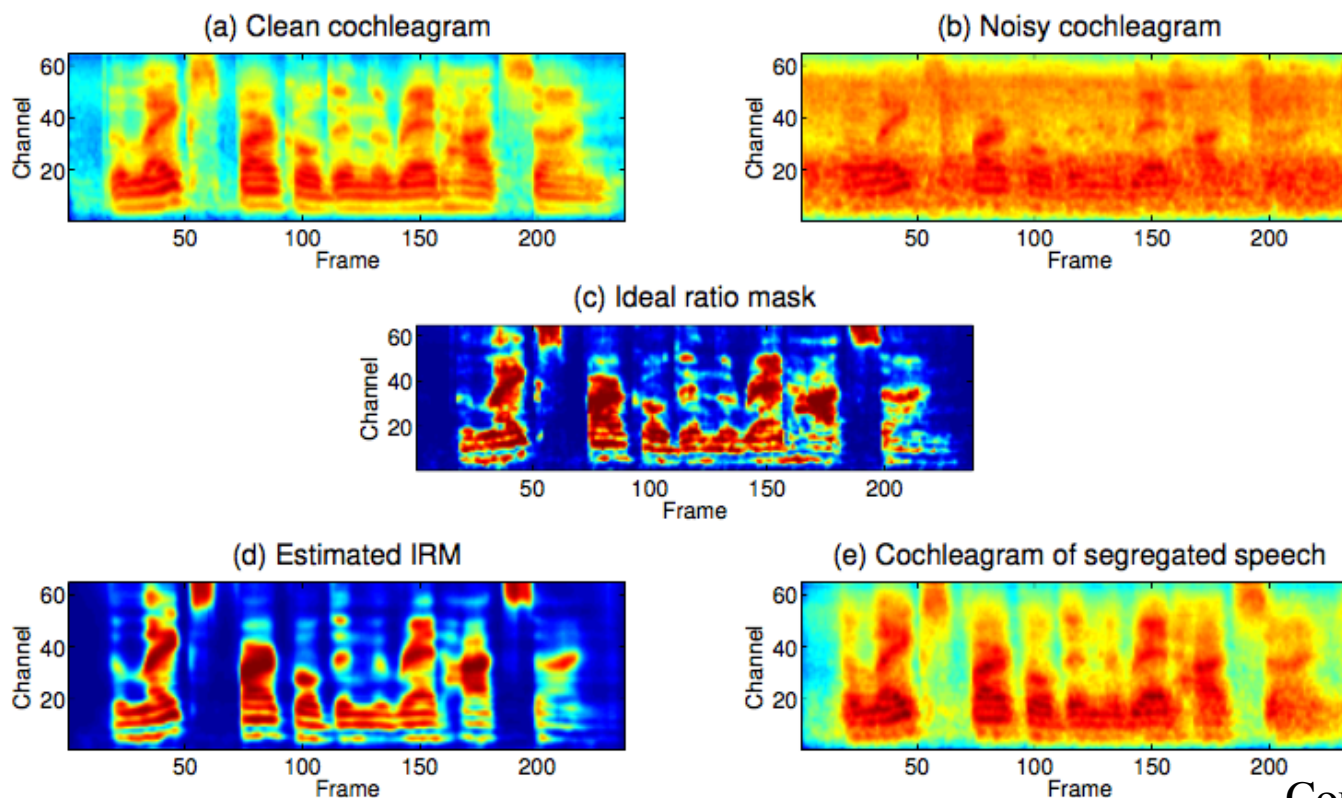
Grouping



We can group sounds into foreground (left)
and background (right)

Machine Learning for Separation

- We can treat the problem of separating signals (e.g. speech from noise) as a ML problem



Courtesy Jitong Chen

Neural Network Speech Separation

- Basic idea: for each Time-Frequency (TF) unit, predict one of these:
 - Whether it is speech dominated or noise dominated (classification)
 - Target: Ideal Binary Mask
 - What the ratio of speech to noise energy is
 - Target: Ideal Ratio Mask
 - What the complex magnitude and phase is
 - Target Complex Ratio Mask (Donaldson 2016)

Chen 2017: Robust Ratio Mask estimation

- Two basic ideas (oversimplified):
 - In order to make the estimator more robust to noise, speaker, and channel variation, perturb the training set in various ways
 - Use LSTMs rather than MLP-DNNs to track long-term speaker characteristics

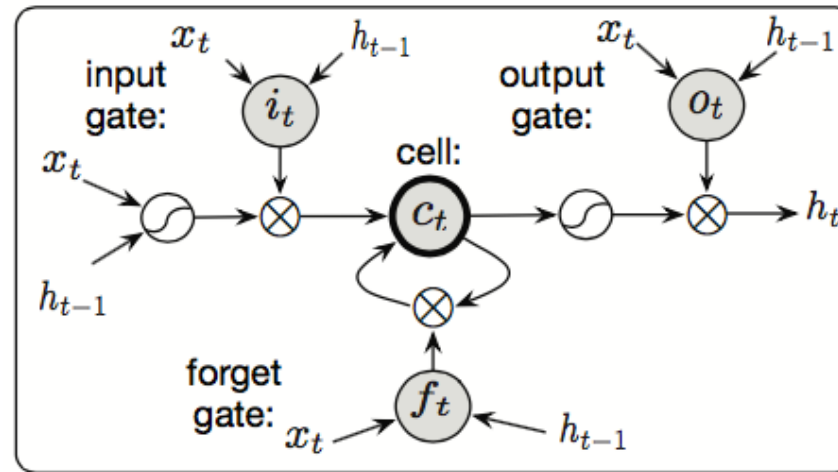


Figure 6.2: Diagram of an LSTM block with three gates and a memory cell.

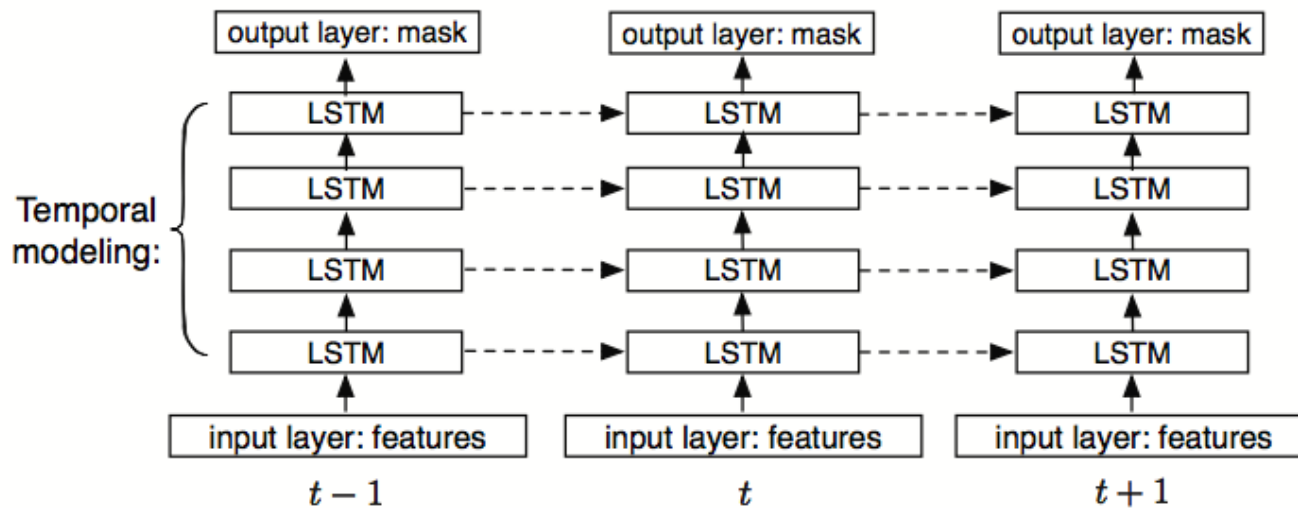


Figure 6.3: Diagram of the proposed system. Four stacked LSTM layers are used to model temporal dynamics of speech. Three time steps are shown here.

Sound Demo 1 (Artificially Mixed Noisy Speech)

- Test set: unseen male and female speakers are mixed with unseen noises at -2 dB, 0 dB and 5 dB SNR.

Unprocessed (**-2 dB**):



Processed:



Clean:



Unprocessed (**0 dB**):



Processed:



Clean:



Unprocessed (**5 dB**):



Processed:



Clean:



Sound Demo 2 (Real Recording in Apartment)

- Unprocessed:



- Processed speech by a **speaker-, noise- and channel-independent** model:



Audio-Technica
microphone

Sound Demo 3 (Real Recording in Apartment)

- Unprocessed:



- Processed speech by a **speaker-, noise- and channel-independent** model:



Macbook built-in microphone

Sound Demo 4 (Real Recording in Restaurant)

- Unprocessed:



- Processed speech by a **speaker-, noise- and channel-independent** model:



Audio-Technica
microphone

Sound Demo 5 (Real Recording in Restaurant)

- Unprocessed:



- Processed speech by a **speaker-, noise- and channel-independent** model:



Audio-Technica
microphone

Courtesy Hong Chen

Unit 3: Application Areas

3c. Audition & Music Retrieval

Music Retrieval & Indexing

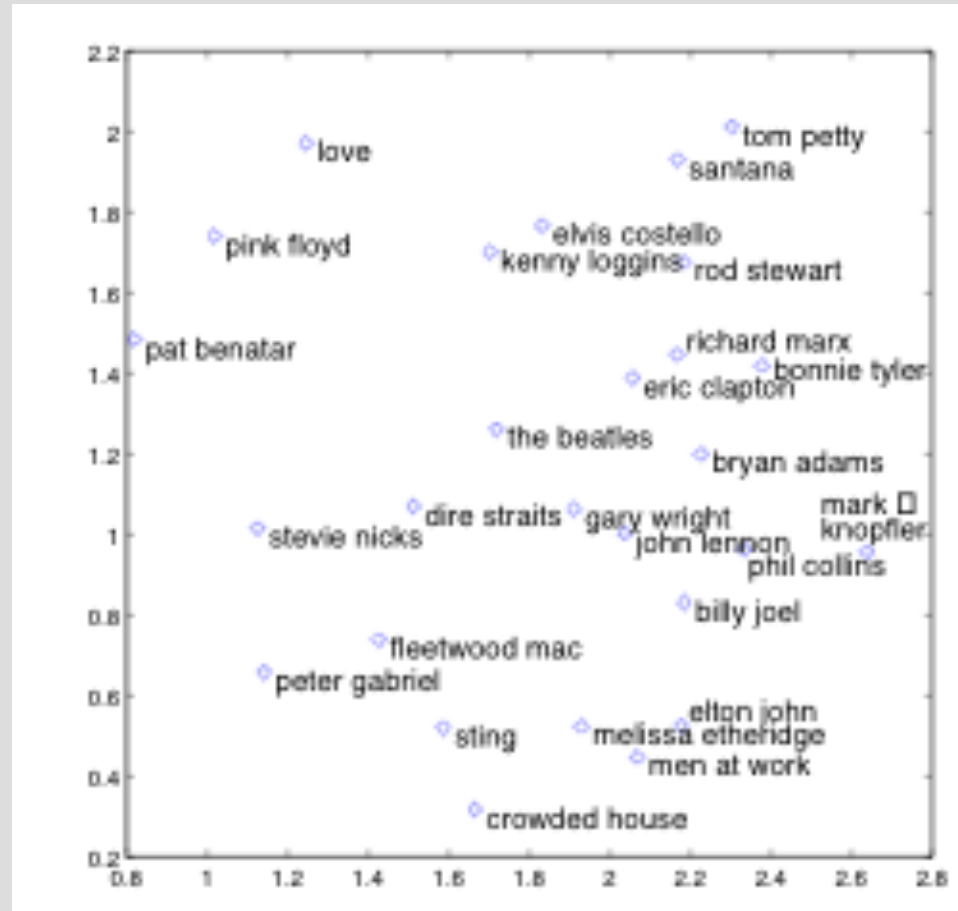
- Lots of music out there nowadays
- How do you know
 - a) what you might like to listen to?
 - b) how to find a song where you don't know the title?
- AI / machine learning techniques can be applied to help solve this problem

Artist similarity

- People have difficulty trying to describe new artists; experts can do it to some degree:
 - Jeff Buckley is “Van Morrison meets Led Zeppelin” but is “more folkie” and influenced by “lounge jazz” -- All Music Guide (Berenzweig et al 2003)
- Other metrics:
 - User surveys (which artist these 10 is this one like...)
 - User playlists on the web
 - User music collections (Napster in the old days)

Artist similarity & machine learning

- Can we automate this type of recommendation service?
- Idea: find some feature representation of music; learn to classify based on feature representation



Ellis et al 2002

Anchor Spaces Algorithm (Berenzweig et al 2003)

- Pick 14 dimensions of style space
 - 12 genres (rock, punk, country...) + male/female, low/high fidelity
- Train a neural network on canonical examples of each dimension
 - Use many small snippets of songs
 - Binary decision: is this rock? (Yes/No)
- A new music snippet evaluated with each classifier ($P(\text{genre}|\text{snippet})$) gives rise to a point in 14-dimensional space

Anchor Spaces Algorithm (Berenzweig et al 2003)

- For every artist, find the 14-dimensional points for his/her songs
- Train a Mixture of Gaussians for this artist
 - This gives the likelihood of snippet space for this artist
 - Will have one MoG for each artist
- When a new song comes in, compare its snippets to all of the MoGs
 - MoG that gives data highest likelihood is output
 - Choose 1-of-25 task: 46 to 62% correct

Query by Humming

(Pardo et al 2004)

- Idea: you have a large database of music
 - Want to find a song but don't have title
- Picking pitches out of music difficult
 - But often have MIDI files (e.g., Karayoke)
- Input: user hums tune, match to MIDI representation
 - Problem: user out of tune/wrong key
 - Problem: user doesn't match timing

QBH: Representation

- Pitch representation
 - Do pitch detection
 - Clean it up by matching to nearest pitch
 - Build in a model of how far off someone might be
 - Represent as change in pitch from previous note (3 steps up, 2 steps down)
- Timing representation
 - Represent as change in timing from previous note (longer/shorter/same)

QBH: matching

- Problem then just becomes trying to match these *relative* representations to MIDI representations
- ML used to find ways of matching

