Using Cooperative Ad-hoc Microphone Arrays for ASR

JSALT 2019 School – June 19th, 2019

Maurizio Omologo, Fondazione Bruno Kessler (FBK), Trento, Italy

Mirco Ravanelli, MILA Montreal, Montreal, Canada

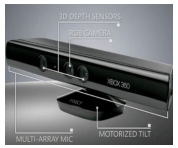
Outline of this introductory lecture

- General goals of our workshop
- Introduction to lectures and labs of today
- Quick tour over some background of digital signal processing
- From speech signal to mel-cepstral coefficients
- Multi-microphone signal processing for distant-speech recognition (DSR)
  - General problems and related tasks
  - CHiME5 and DIRHA
  - Room acoustics and sound propagation with microphone arrays
  - Acoustic impulse responses and image method for data contamination

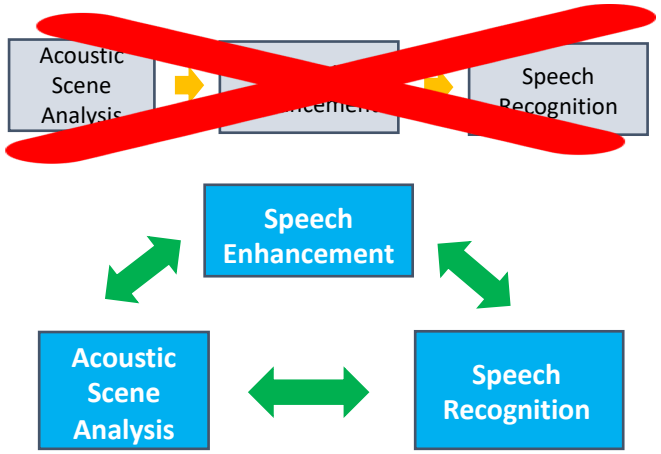
General goals of our workshop

- **Distant Speech Recognition** (DSR) under real-world challenging conditions, in particular by using **ad-hoc microphone arrays** distributed in space
  - Main goal of developing **fully automatic methods to remedy clock drifts**, and of evaluating their impact on speaker activity alignment as well as on DSR performance.
  - Towards **releasing constraints and supervision** on some tasks (such as ground-truth segmentation in CHiME5)

- State-of-the art
  - Recent significant advances thanks to **deep learning**
  - Huge number of works and projects (e.g., **CHiME5** and DIRHA)
  - Yet **limited performance** due to many different factors, e.g., noise, reverberation, overlapping speakers, spontaneous speech. **Clock drift** is a further key factor. Another issue may be **sample loss**



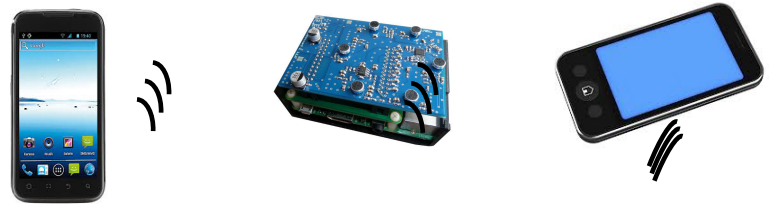

- Other goals of the workshop
  - From fully-supervised to ***semi/un-supervised learning solutions***
  - Explore the use of ***cooperative neural approaches*** for DSR
  - Tackle specific problems introduced by ***ad-hoc microphone arrays***
  - ***Multi-channel speaker diarization*** as pre-processing step
  - Exploit the ***PyTorch-Kaldi*** framework
  - Use of ***realistic multi-channel simulated data*** for training and dev.
  - ***Public distribution*** of some recipes and data after the workshop

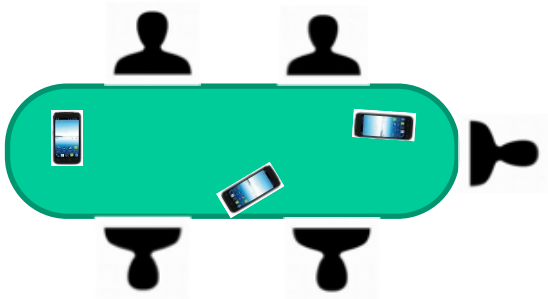


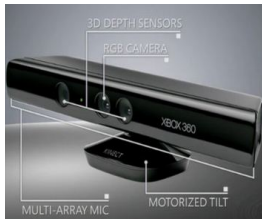
Agenda of the day

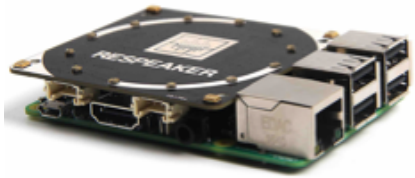
9:00 – 10:30 AM **Multi-microphone signal processing for distant-speech recognition (DSR)** - Maurizio Omologo
10:30 – 10:50 AM Break
10:50 AM – 12:10 PM Cooperative and self-supervised neural frameworks for DSR – Part I - Mirco Ravanelli
12:10 – 1:00 PM Lunch Break
1:00 – 2:00 PM Cooperative and self-supervised neural frameworks for DSR – Part II - Mirco Ravanelli
1.30 – 2.00 PM PyTorch Kaldi - Mirco Ravanelli
2:00 – 3:00 PM Multi-microphone data-sets and tasks – Part I - Maurizio Omologo
3:00 – 3:30 PM Coffee Break
3:30 – 4:30 PM Multi-microphone data-sets and tasks – Part II - Maurizio Omologo
4.30 - 5.00 PM Questions and conclusions

Microphone networks







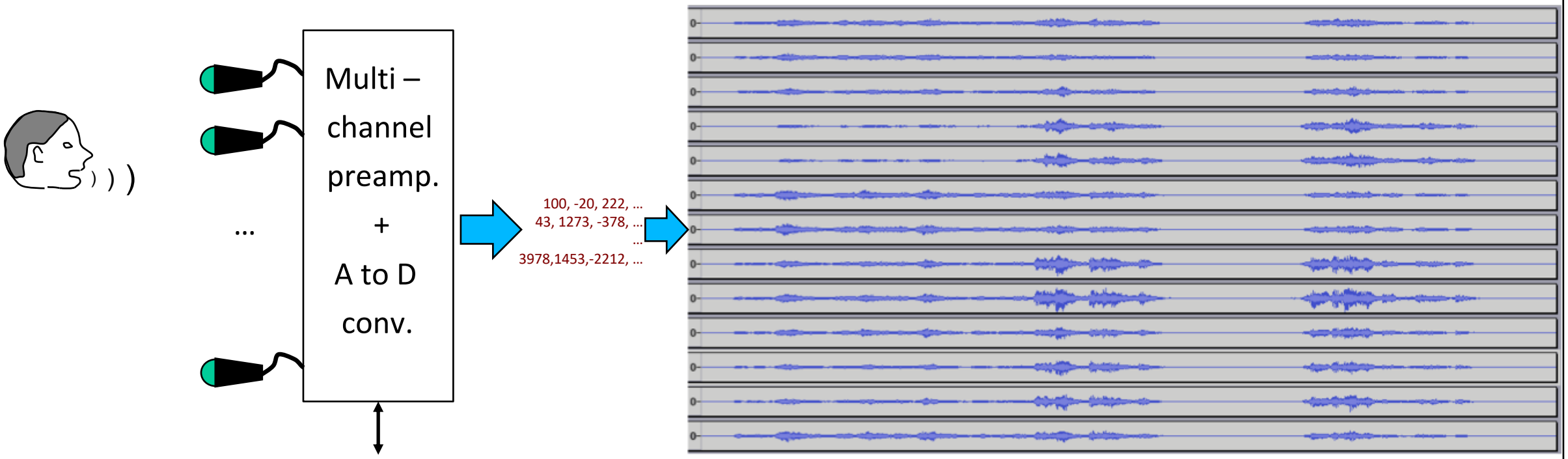


- From traditional stand-alone microphone array devices
- To cabled solutions with distributed microphone arrays
- To networks of ad-hoc microphone arrays and networks

A very quick tour across notions of Digital Signal Processing (DSP)

… we always deal with multiple sequences of numbers.

Let’s see where do they come from, and how to process them



- ✓ **SAMPLE ACCURACY**
- ✓ **NOMINAL SAMPLING FREQUENCY (a common clock shared by all vs independent clocks)**
- ✓ **LOSSLESS AUDIO FORMAT !!!**

Multi-microphone signal processing for distant-speech recognition (DSR)

JSALT 2019 School – June 19th, 2019

Maurizio Omologo, Fondazione Bruno Kessler (FBK), Trento, Italy

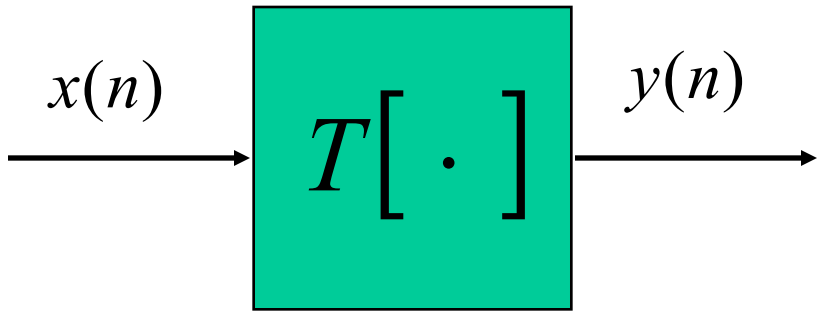
A quick tour* across notions of Digital Signal Processing (DSP)

*) very limited, without the introduction and application of z-transform

Suggested readings on these topics:

- - A.V. Oppenheim and R.W. Schafer, *Discrete-time Signal Processing*, 3-rd edition , Prentice Hall-Pearson, 2010.
- - L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, Englewood Cliffs, 1978.
- - L.R. Rabiner and R.W. Schafer, *Introduction to Digital Speech Processing*, Foundations and Trends in Signal Processing, Vol. 1, 2007. - available online.

Discrete-time systems



$$y(n) = T[x(n)]$$

A discrete-time system can be seen as a process **T** that produces an ***unambiguous transformation of an input sequence*** $x(n)$ in an output sequence $y(n)$

- Systems (not only discrete ones) can be connected one each other as **cascade**, **parallel**, or **cascade/parallel** combinations
- Another important and common way to combine is ***feedback***
- The main properties of a system are: **memory, invertibility, linearity, time-invariance, causality, stability.**

Impulse Response – Convolution sum

- $h(n,k)$ is defined as **impulse response** (or response to the impulse occurring at $n=k$) of the transformation T
- The impulse response is a **complete characterization of the properties of a specific linear system**
- Given a Linear Time-Invariant (LTI) system, one has the following property: $y(n-k)=T\left[x(n-k)\right]$ where k denotes a non-null integer.
- If n refers to time, one deals with **time-invariance** and can represent it as follows: $h(n,k)=T\left[\delta(n-k)\right]=h(n-k)$ where $h(n)$ denotes the impulse response of LTI system
- Hence, the system output can be given as follows:

$$y(n)=T\left[x(n)\right]=\sum_{k=-\infty}^{+\infty}x(k)h(n-k)$$

- One can say that $y(n)$ is the ***convolution*** of $x(n)$ and $h(n)$. It is called as ***convolution sum*** and denoted as: $y(n)=x(n)*h(n)$

Linear constant-coefficient difference equations

- An important subclass of linear time-invariant systems consists of those systems for which the input and the output satisfy a difference equation of the form:

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{r=0}^{M} b_r x(n-r)$$

- Note: in general they may be **non-causal systems** (see $N=2, M=0, a_0=0 \Rightarrow a_2 y(n-2) + a_1 y(n-1) = b_0 x(n)$)
- …In the following **we will always address causal systems**; for these systems, one can describe the **explicit input-output relationship** as a recursive formula, i.e., in the form:

$$y(n) = -\sum_{k=1}^{N} \frac{a_k}{a_0} y(n-k) + x(n) + \sum_{r=1}^{M} \frac{b_r}{a_0} x(n-r)$$

- LTI systems are characterized by the fact that the steady-state response, to a **sinusoidal input at a given frequency**, is sinusoidal at the same frequency, with **amplitude** and **phase** depending on system properties

- Hence, one can consider a complex sinusoidal input $x(n) = e^{j\omega n}$ $-\infty < n < \infty$, characterized by the frequency variable ω, and an impulse response $h(n)$ of the system. Consequently, the system output is:

$$y(n) = h(n) * x(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) = \sum_{k=-\infty}^{\infty} h(k)e^{j\omega(n-k)} =$$

$$= e^{j\omega n} \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k} = H(e^{j\omega})e^{j\omega n} \quad \text{with } H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k}$$

- In other words, the quantity $H\left(e^{j\omega}\right)=\sum_{k=-\infty}^{\infty}h(k)e^{-j\omega k}$ represents the **frequency response** of a system having $h(n)$ as impulse response

- This quantity describes the change in complex amplitude of a complex exponential input signal as a function of the frequency ω

- $H\left(e^{j\omega}\right)$ is complex and can be expressed in terms of its real and imaginary part $H\left(e^{j\omega}\right)=H_R\left(e^{j\omega}\right)+jH_I\left(e^{j\omega}\right)$ or in terms of magnitude and phase $H\left(e^{j\omega}\right)=\left|H\left(e^{j\omega}\right)\right|e^{j\arg\left[H\left(e^{j\omega}\right)\right]}$

- Note that $H\left(e^{j\omega}\right)$ is periodic with period 2π (since $e^{j\omega k}=e^{j(\omega+2\pi)k}$)

Frequency-domain representation of discrete-time signals and systems – Parseval's relation

- Note also that from the quantity $H\left(e^{j\omega}\right)=\sum_{k=-\infty}^{\infty}h(k)e^{-j\omega k}$ one can obtain the impulse response by applying the inverse Fourier transform integral

$$h(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}H\left(e^{j\omega}\right)e^{j\omega n}d\omega$$

- The two relationships form a **Fourier representation** for the sequence $h(n)$
- A sufficient condition for the existence of the Fourier transform is that the given sequence is **absolutely summable** which corresponds to have: $\sum_{k=-\infty}^{\infty}\left|h(k)\right|<\infty$
- Given two complex sequences $x(n)$ and $y(n)$, and, under some specific conditions, we have the Parseval's relation:

$$\sum_{n=-\infty}^{+\infty}x(n)y^{*}(n)=\frac{1}{2\pi}\int_{-\pi}^{+\pi}X(e^{j\omega})Y^{*}(e^{j\omega})d\omega \Rightarrow \sum_{n=-\infty}^{+\infty}x^{2}(n)=\frac{1}{2\pi}\int_{-\pi}^{+\pi}\left|X(e^{j\omega})\right|^{2}d\omega$$

Discrete Fourier Transform - DFT

$$X(k) = \begin{cases} \sum_{n=0}^{N-1} x(n) W_N^{kn} & \text{for } 0 \le k \le N\text{-}1 \\ 0 & \text{elsewhere} \end{cases}$$

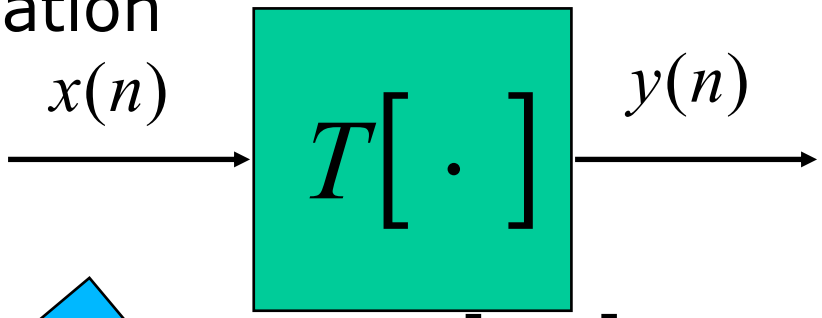
ANALYSIS EQUATION

$$x(n) = \begin{cases} \frac{1}{\mathrm{N}} \sum_{k=0}^{N-1} X(k) W_N^{-kn} & \text{for } 0 \le n \le N\text{-}1 \\ 0 & \text{elsewhere} \end{cases}$$

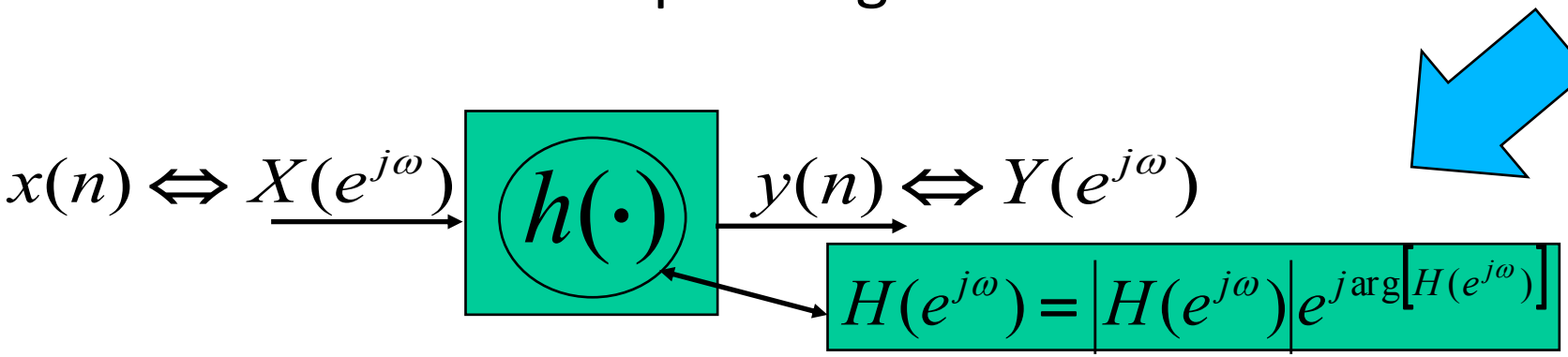
SYNTHESIS EQUATION

$$W_N = e^{-j(2\pi/N)}$$

Let us recall the linear time-invariant transformation



to introduce the concept of digital filter:



where in general: $y(n) = -\sum_{k=1}^{N} \frac{a_k}{a_0} y(n-k) + x(n) + \sum_{r=1}^{M} \frac{b_r}{a_0} x(n-r)$

There are two main classes of digital filters:
Finite Impulse Response (FIR) and **Infinite Impulse Response** (IIR)

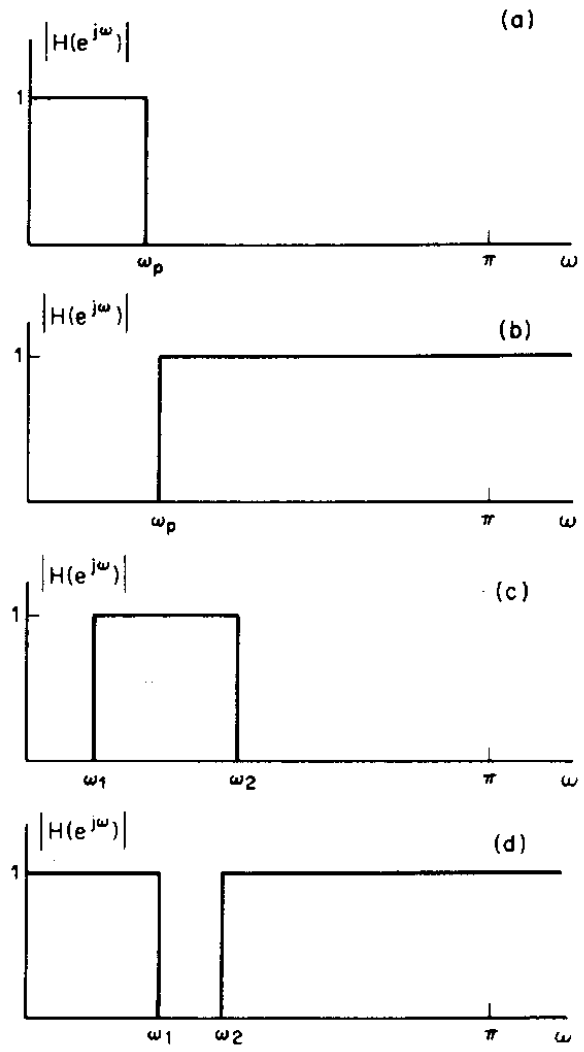
Some categories of digital filters

a) ***Low-Pass (L.P.)***

b) ***High-Pass (H.P.)***

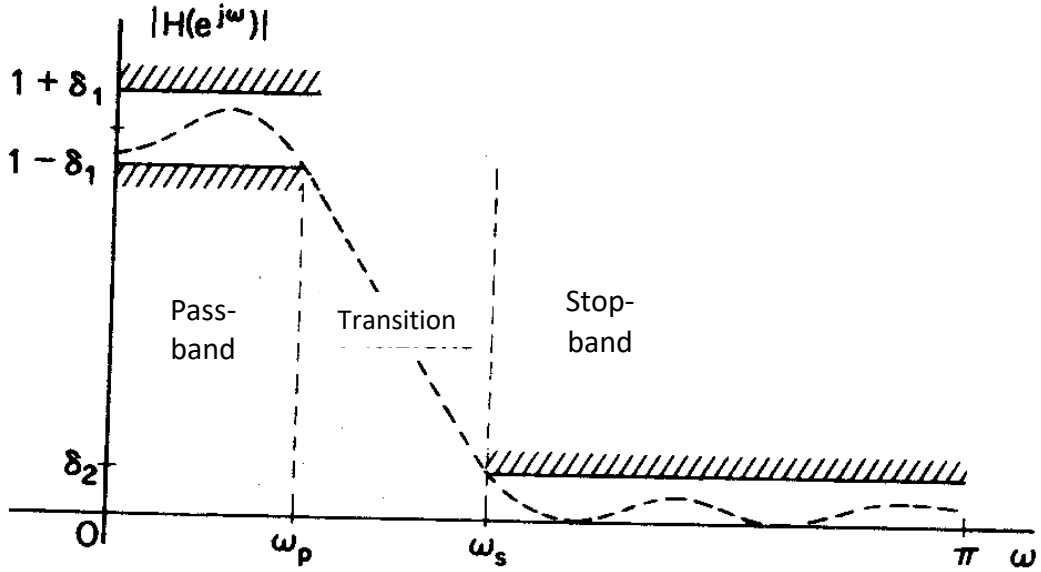
c) ***Band-Pass (B.P.)***

d) ***Band-Stop (B.S.)***



Filter design: limits of tolerable approximation error for an ideal low-pass filter

We have:



where:

$$1-\delta_1 \le \left|\mathrm{H}(\mathrm{e}^{\mathrm{j}\omega})\right| \le 1+\delta_1 \quad \text{for } |\omega| \le \omega_p$$

$$\left|\mathrm{H}(\mathrm{e}^{\mathrm{j}\omega})\right| \le \delta_2 \quad \text{for } \omega_s \le |\omega| \le \pi$$

and ω_p , ω_s are referred to as **cutoff frequencies**

FIR filters – phase linearity

- One of the main advantages of a FIR filter over the IIR filters is given by the fact that in the former case one can obtain ***a linear phase response***, which is a very important property, in general not provided by IIR filters
- Let us consider the generic frequency response of a FIR filter:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n}$$

- The filter is fully described by the finite-duration sequence $h(n)$, that is by the *N* samples of the corresponding Fourier transform
- For the given impulse response we have that $h(n) = h(N-1-n)$

  if and only if the filter has a linear phase response

Short-time Fourier analysis

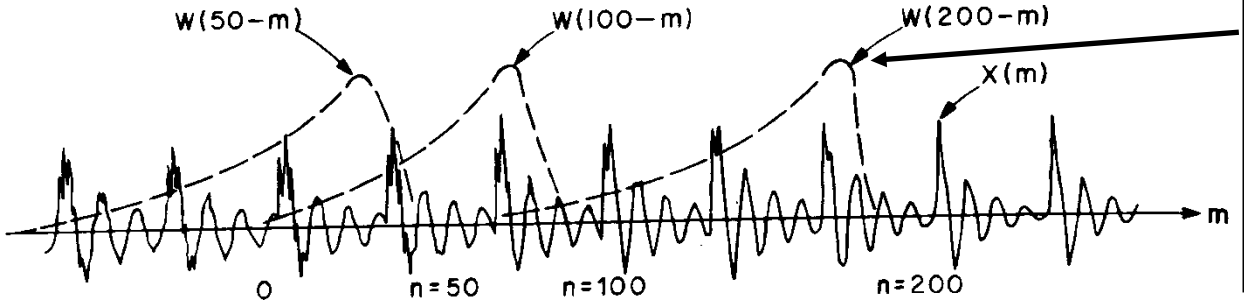
- In many digital signal processing applications, in general one has to deal with waveforms having ***time-varying*** properties; hence, we are motivated to introduce the concept of ***time-dependent Fourier representation.*** In fact, the traditional concept of Fourier transform can not be used here!
- The time-dependent Fourier transform can be simply defined as:

$$X_n\left(e^{j\omega}\right) = \sum_{m=-\infty}^{\infty} w(n-m)x(m)e^{-j\omega m}$$

Hence, at the time-instant n the real window $w(n\text{-}m)$ sequence determines the portion of the input signal $x(\,)$ to use when computing the Fourier transform.

Note: we have two variables!

Short-time Fourier analysis



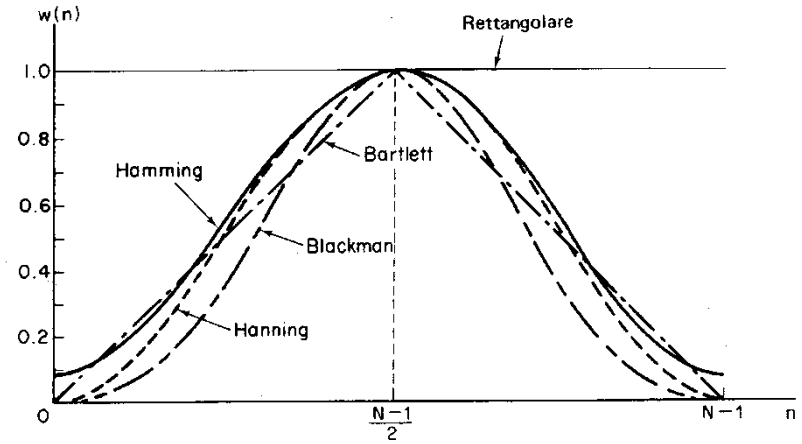
In other words, the window plays the role of emphasizing the samples around *n* and de-emphasizing the samples far from it.

$$\Rightarrow X_n\left(e^{j\omega}\right)=e^{-j\omega n}\sum_{m=-\infty}^{\infty}x(n-m)w(m)e^{j\omega m}=e^{-j\omega n}\widetilde{X}_n\left(e^{j\omega}\right)$$

- The latter expression is alternative to the former one, which means that the equation can be interpreted in two ways:
  - **For fixed** *n* it is the *normal* **Fourier** transform of the sequence $w(n-m)x(m)$
  - **For fixed** ω it is a convolution sum, that is a **linear filtering**:

$$\widetilde{X}_n\left(e^{j\omega}\right)=h(n)*x(n)=\left(w(n)e^{j\omega n}\right)*x(n)\Leftrightarrow X_n\left(e^{j\omega}\right)=\left(x(n)e^{-j\omega n}\right)*w(n)$$

Properties of windows and main lobe width



a) Rectangular ($4\pi / L$, -21 dB)

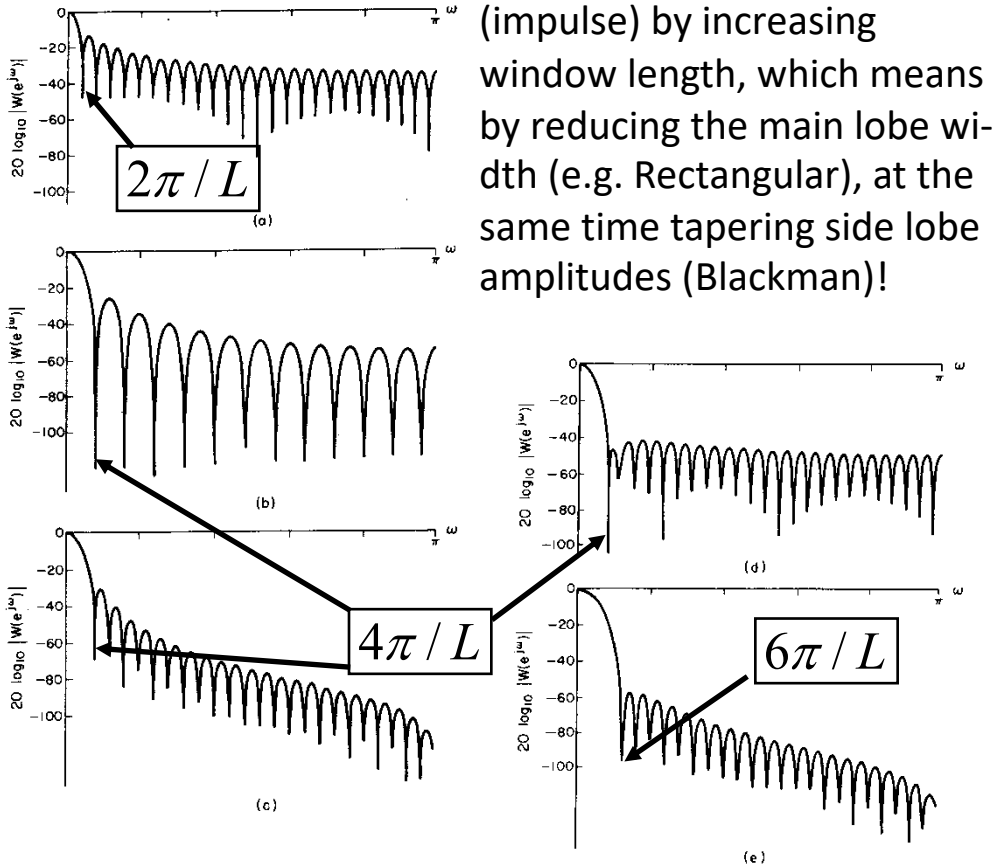
b) Triangular ($8\pi / L$, -25 dB)

c) Hanning ($8\pi / L$,-44 dB)

d) Hamming ($8\pi / L$,-53 dB)

e) Blackman ($12\pi / L$,-74 dB)

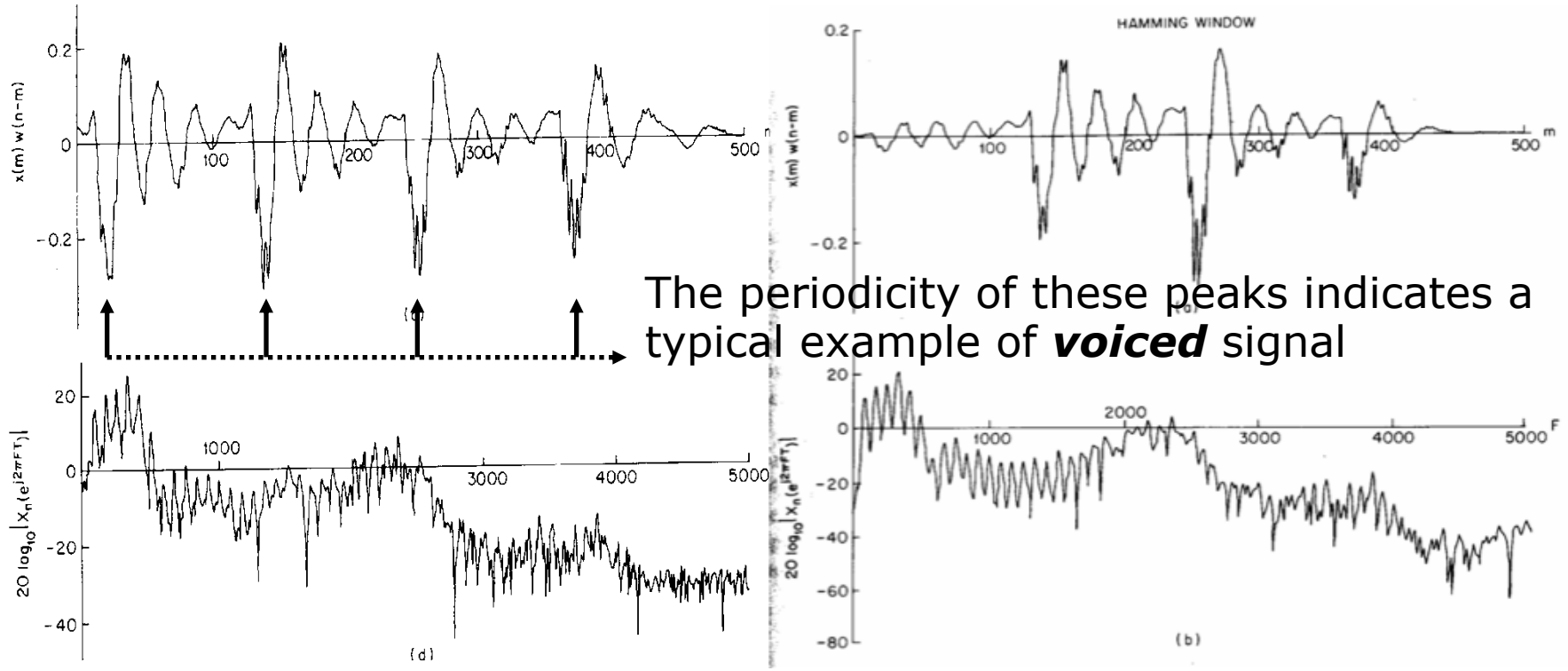
(main lobe width, max. height of side lobes)



NOTE: we would approach the ideal unit-sample function (impulse) by increasing window length, which means by reducing the main lobe width (e.g. Rectangular), at the same time tapering side lobe amplitudes (Blackman)!

NOTE: the main lobe comprises also negative frequencies till the first zero!

Short-time Fourier analysis: effects of windowing



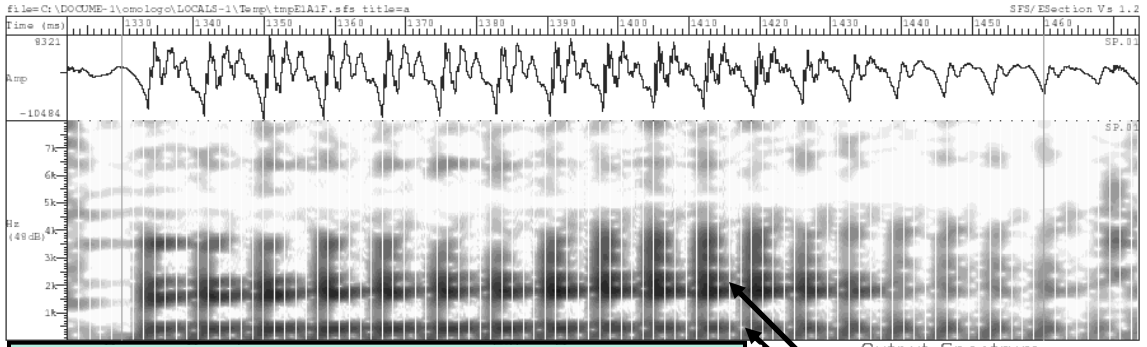
The periodicity of these peaks indicates a typical example of ***voiced*** signal

Analysis of a speech segment of duration equal to 50*ms* using *rectangular* and *Hamming* windows

Spectrogram: a very common tool for speech analysis

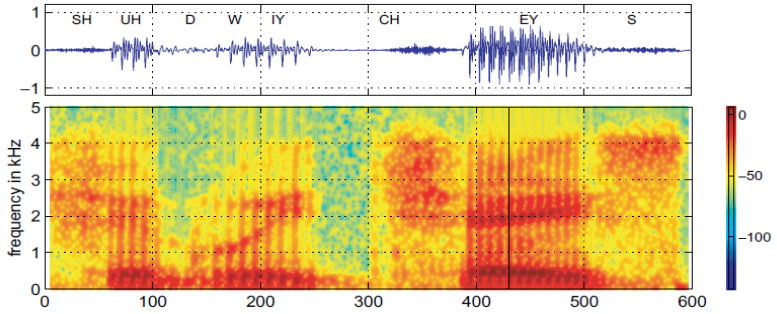
Since the 1940s, the ***Spectrogram*** has been a basic tool for gaining understanding of how speech is produced and how phonetic information is encoded in it

- It consists in a gray-scale or a color-mapped image on a time (x axis)-frequency (y axis) plane.
- The gray or color intensity denotes the magnitude of the Short-Time Fourier Transform of the given signal segment for a given time instant and frequency
- Examples of **wide-band spectrogram**

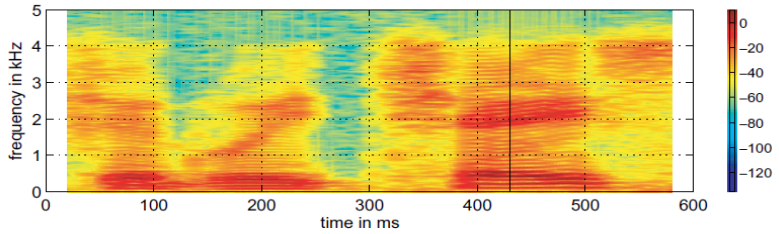


Wideband: STFT analysis done with a short window (e.g. 15 ms) – i.e. a broad bandwidth of the filtering in frequency

Narrowband: larger window size (e.g. 50 ms)



… and **narrow-band** one



Introduction to mel cepstral coefficients

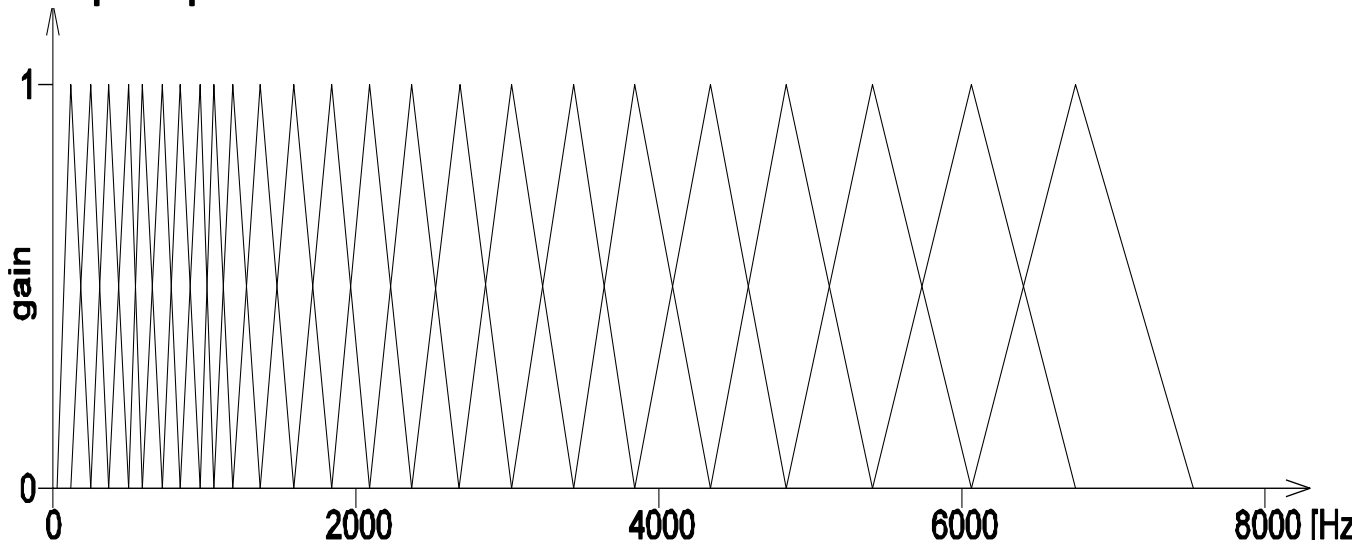
- ***Mel cepstral coefficients*** is an example of use of Short-time Fourier Transform
- They are a commonly used set of acoustic features for speech recognition
- They were introduced in 1982 by Davis-Mermelstein, and are based on a frequency warping justified at perceptual level
- Let us address the description of the cepstrum computation. It starts from the inverse DFT applied to the logarithm of the magnitude signal spectrum of $x()$:

$$\hat{c}_a(n) = \frac{1}{N}\sum_{k=0}^{N-1} \log\left|X_a(k)\right| e^{j2\pi kn/N}$$

- In this case a grid of N bins in frequency is used, with adjacent bins equispaced (of $2\pi/N$) between 0 and 2π
- In order to compute the mel cepstral coefficient vector a ***bank of mel-spaced filters*** is introduced

Mel cepstral coefficients: the concept of filter-bank

- First of all note that with the term *filter-bank* one refers to a processing technique that corresponds only roughly to the ***traditional*** filter-bank processing
- In this case, in fact, a ***triangular filter*** is *realized* as follows: "*weighted average of magnitudes of the DFT, computed on a set of frequency bins that are around a given frequency, which corresponds to the center of the triangular filter band.*"
- Example of mel-equispaced filter-bank in the case of 16 kHz sampling frequency

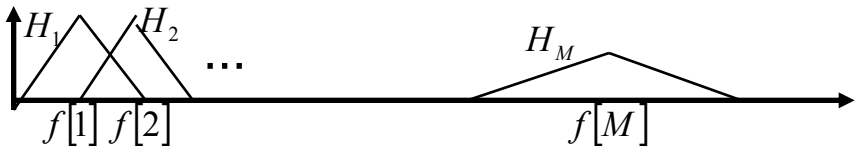


- The bandwidth of each triangular filter increases with frequency, and the central frequencies of the filters are logarithmically spaced (above 1000 Hz)

Filter-bank for computation of the mel cepstral coefficients

The definition of generic m-th triangular filter H_m is described by the following relationships (where k represents the DFT index):

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & f[m-1] \le k \le f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases}$$



Computation of mel cepstral coefficients

First of all, the output (log-energy) of each triangular filter is computed (by means of a weighted average of the DFT magnitude values)

$$S[m] = \log\left[\sum_{k=0}^{N-1} |X_a[k]| H_m[k]\right]$$

NOTE1: In the literature, some authors include $|X_a[k]|^2$ in the sum, others include magnitude (as here). Other alternative: log on each component of the sum.

Then, mel-cepstral coefficients are derived as follows:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left[\pi n (m + {}^1/_2)/M\right]$$

$$1 \le n \le N_c$$

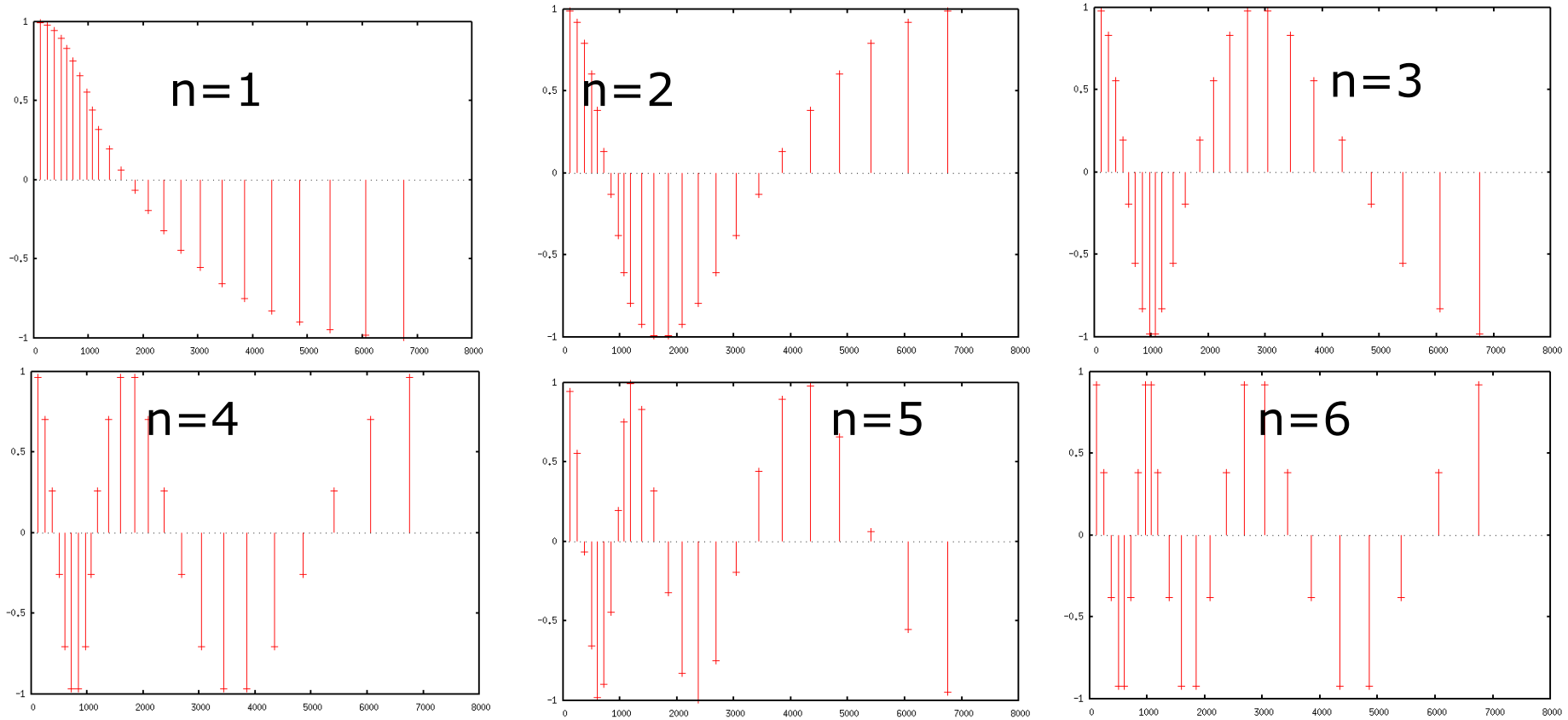
NOTE2: We have a DCT instead of an inverse DFT, since $S[m]$ is an even function

Note that the *M* components of the vector $S[m]$ are generally highly correlated one each other, while in $c[n]$ that correlation is definitely reduced.

Typical values of *M* are from 24 to 40. The size N_c of the vector of mel cepstral coefficients is typically in the range between 8 and 13.

Interpretation of the computation of mel cepstral coefficients

These figures show the various functions used to compute the first six mel cepstral coefficients: note the relationship between **first mel coefficient** and **spectral slope**



Multi-microphone signal processing for distant-speech recognition

Preliminary aspects

- Huge literature on these two fields and on related scientific topics among which:
  - Speech recognition, deep learning, acoustic modeling, language modeling, etc.
  - Sound source localization, speech enhancement (e.g., beamforming, noise subtraction, dereverberation, source separation), echo cancellation, acoustic event/speech activity detection, speaker diarization, data contamination/ augmentation, etc.
  - Ad-hoc microphone array processing, synchronization, clock drift mitigation, etc.
- We will mainly focus on multi-microphone signal processing, and refer to CHiME5, as most actual and complex experimental framework under which research is conducted
  - As for CHiME5, see also past challenges, starting from Pascal (2006) and CHiME (2008), and pdf slides available online, of a recent keynote of Jon Barker
  - In CHiME5: conversational speech transcription in domestic environments based on multi-microphone devices distributed in space
  - From the last challenge, many remaining open issues, and need to jointly progress on acoustic scene analysis, speaker diarization, and robust speech recognition
- See also other past projects, such as EC-DICIT, EC-DIRHA, EC-AMI/AMIDA, EC-CHIL

CHiME5 Challenge

- Tracks: Single array (specifying the reference array), Multiple arrays (using all)
- Results:
  - Baseline performance around 92% WER, with GMM, and 81% with DNN
  - Output of the challenge: best system performance around 45-50% WER
  - More recently, best systems around 40% WER
  - So far, not evident a clear improvement when using multiple arrays
- NOTE: currently manual segmentation (i.e., start-end of each utterance) is exploited
- Besides the development of automatic segmentation, other critical issues to tackle are:
  - overlapped speech (more than 20-25%)
  - highly conversational spontaneous speech
  - unstationary noisy events
  - distant-speech, attenuation and reverberation
  - possibly limited training material size
  - speech often reflected (no direct-path)
  - cross-device synchronization, clock drift, sample drop
  - no knowledge about device positions

CHiME5 Challenge: a variety of approaches and components

- In general, very complex system architectures, both in the front-end multi-microphone processing and in the back-end . Examples of techniques and components:
  - Weighted Prediction Error (WPE) for dereverberation
  - Beamforming for source extraction (e.g., use of BeamformIt, MVDR, NN-based)
  - Speaker-adaptive systems
  - Data augmentation
  - Room simulators
  - Guided Source Separation
  - Source Activity Detector
  - Channel Selection
  - TDNNs, Convolutional, recurrent (LSTM, BiLSTM, …) deep architectures for back-end
  - Speaker embeddings
  - Speaker aware acoustic model architectures
  - Multiple systems, multi-pass, rescoring, ROVER for system combination
- For more details, see results and papers available in the challenge website and in http://spandh.dcs.shef.ac.uk/chime_workshop/programme.html

Past projects: DIRHA

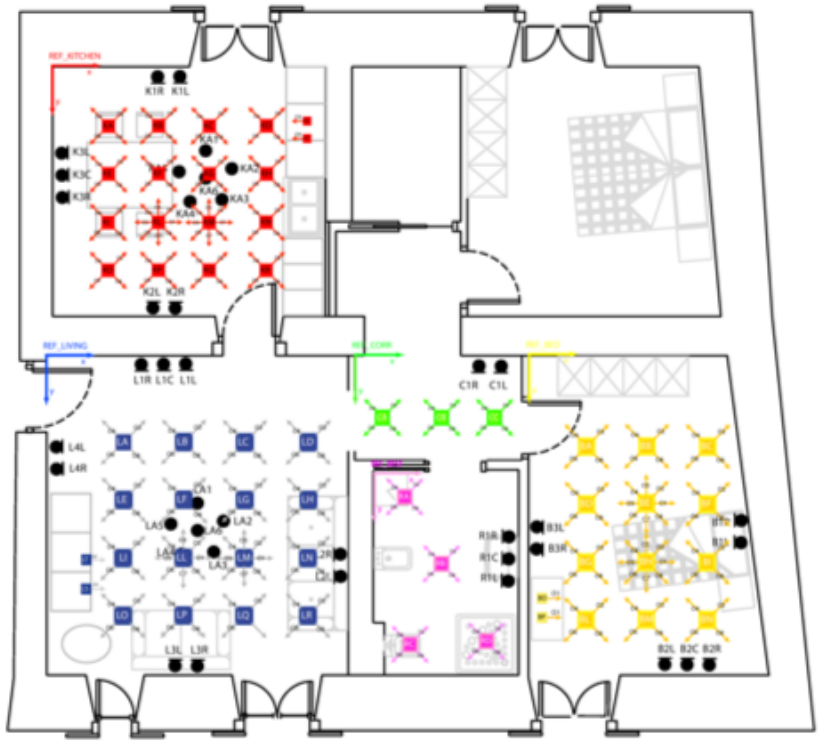
Acoustic scene analysis

Distant-speech interaction

Voice-enabled home automation

Funded by EC – FP7
Collab. Project – STREP
ICT-2011-7
Language technologies
From 2012 to 2014

- Distributed microphone network
- Always listening system, no-push-to-talk activation
- Multi-room, multi-speaker/sound sources
- Robust to typical conditions of domestic contexts
- Reduced invasiveness, no-cabling in the targeted final application
- Easy portability to other languages



Selected topics: focus on some multi-microphone processing tasks

- In the afternoon we will see examples related to:
  - Data contamination for generation of artificial data
  - Sound source localization and extraction of localization cues
  - *Beamforming*
  - *Clock drift analysis*
  - *Sample drop detection*

- Propedeutic to this, let's start with an introduction to some basic concepts concerning room acoustics and sound propagation, with microphone arrays

Microphone arrays

Microphone arrays are multichannel acquisition devices that allow sampling acoustic fields:

- in time (synchronously, except for some cases such as ad-hoc arrays)
- in space (with proper geometry)

By processing the signals of a microphone array it is possible to:

- change the directivity of sound acquisition (beamforming)
- apply spatial/temporal filtering
- selectively pick-up and enhance the desired signal
- cancel or attenuate undesired disturbances
- detect and localize acoustic sources

Most common arrays are linear or harmonic
For sound source localization tasks, 2D and 3D geometries are essential (e.g., T-shaped, L-shaped, circular, spherical, etc.)

Microphone arrays: near-field vs far-field sources

- **Far-field** source ⟺ Propagation of sound as a plane wave
- **Near-field** source ⟺ Propagation of sound as a spherical wave

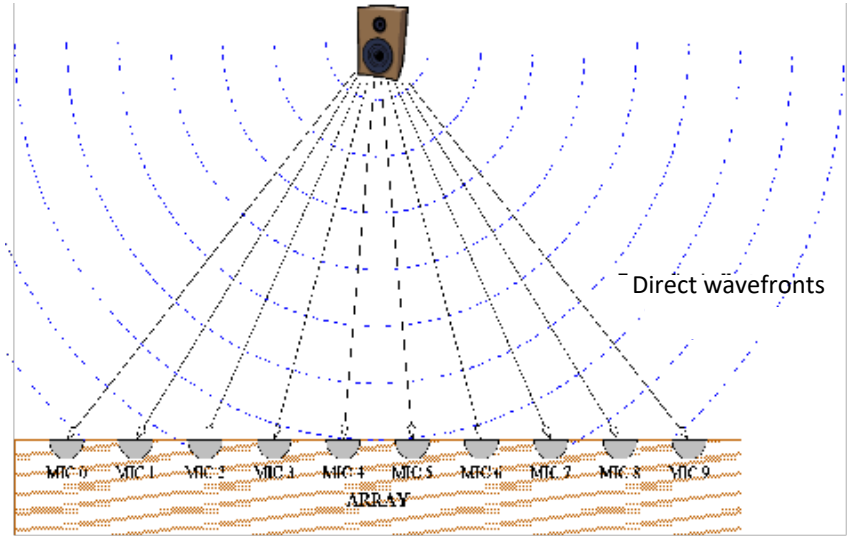
A source can be considered to be in the far-field if

$$r > \frac{2L^2}{\lambda}$$

where: r is the distance to the array,

L is the length of the array, and

λ is the wavelength of the arriving wave.

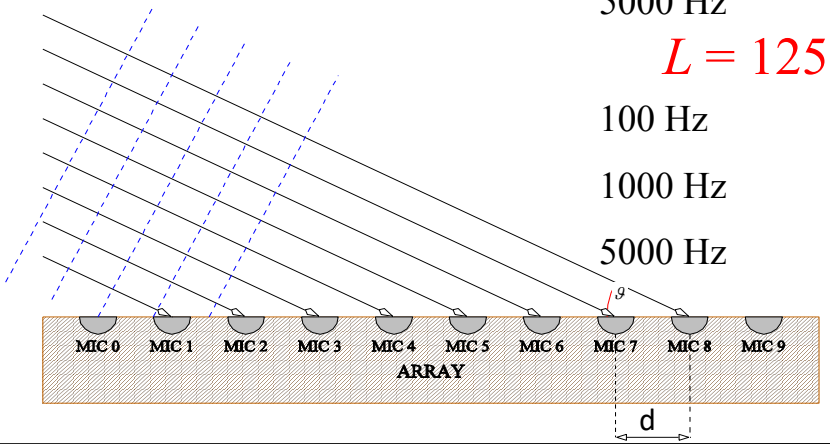


$L = 25\ cm$

$f =$	$r > \sim$
100 Hz	3.7 cm
1000 Hz	37 cm
5000 Hz	1.84 m

$L = 125\ cm$

100 Hz	92 cm
1000 Hz	9.2 m
5000 Hz	46 m



Far, near vs very-near field conditions

- **Far field**
  - Plane wave approximation at the array
  - Source position vs wavefront arrival angle
  - Inverse distance law for sound pressure
  - Examples: large auditoria, outdoor
- **Near field**
  - Spherical wavefront at the array
  - Inverse square law effects
  - Examples: office, large room
- **Very near field**
  - Non-omnidirectional radiation pattern
  - Dominant inverse square law effects
  - Examples: small room, car, aircraft cockpit

Noise field competing with speech in an indoor context

In general the noise field at the sensor results from contribution of different noise sources (of unknown number, position and characteristics).
We can distinguish among the following characteristics:

o Additive	***vs***	o Convolutional
o Coherent field		o Diffuse field
o Point source		o Spatially distributed source
o Narrowband		o Wideband
o Uncorrelated		o Correlated (with speech)
o Stationary (in time and space)		o Non-stationary
o Known		o Unknown

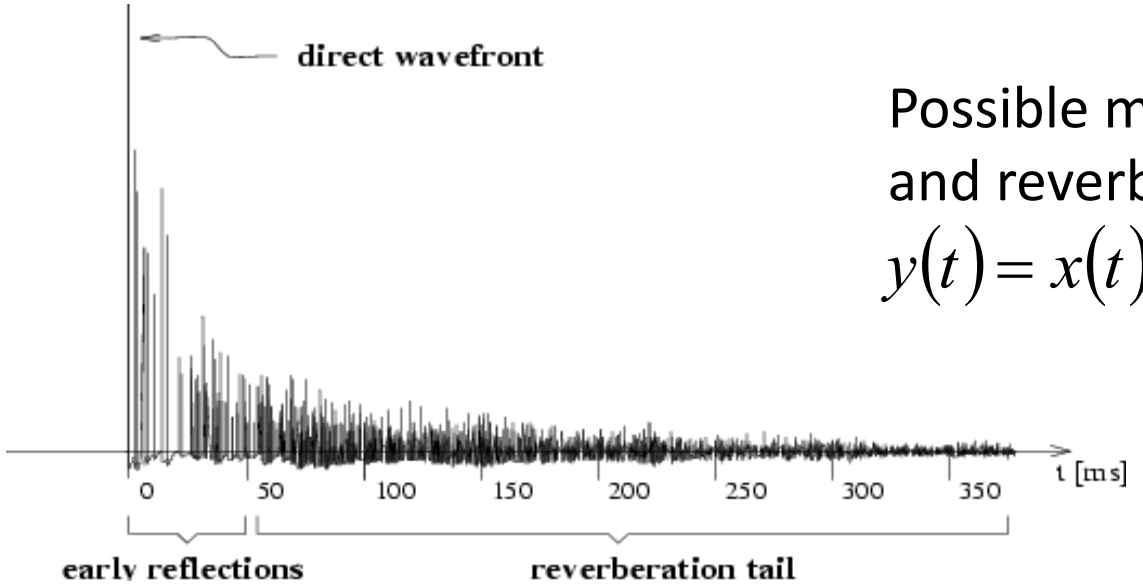
The complexity of tasks such as DSR, source separation, sound source localization can depend on the characteristics of this environmental noise and, in general, on the SNR at each microphone

Reverberation - Acoustic Impulse Response

The **reverberation** phenomenon is due to reflections from surfaces and diffusion and diffraction by objects inside the room.

Sound propagation from a source to a microphone can be modeled (in a simplified manner) by means of a FIR filter.

An acoustic ***Impulse Response*** (IR) describes relative amplitude and time delay of **direct-path** wave arrival and of all **reflections**.



Possible model of noisy and reverberated signal:

$$y(t) = x(t) * h(\boldsymbol{s}, t) + n(t)$$

Reverberation time T60: time required for a decay of 60 dB of intensity for a sound abruptly interrupted.

Acoustic signal model

Source at $\mathbf{s} = [s_x, s_y, s_z]$ Microphone M_i at $\mathbf{m}_i = [m_{ix}, m_{iy}, m_{iz}]$

$x(t)$ = source signal $y_i(t)$ = i-th mic signal $T_i = \frac{|\mathbf{s} - \mathbf{m}_i|}{c} = \frac{r_i}{c}$ = propagation time

Considering attenuation and delay of propagation and additive noise, the simplest model of the input signal is:

$$y_i(t) = \frac{1}{r_i} x(t - T_i) + n_i(t)$$

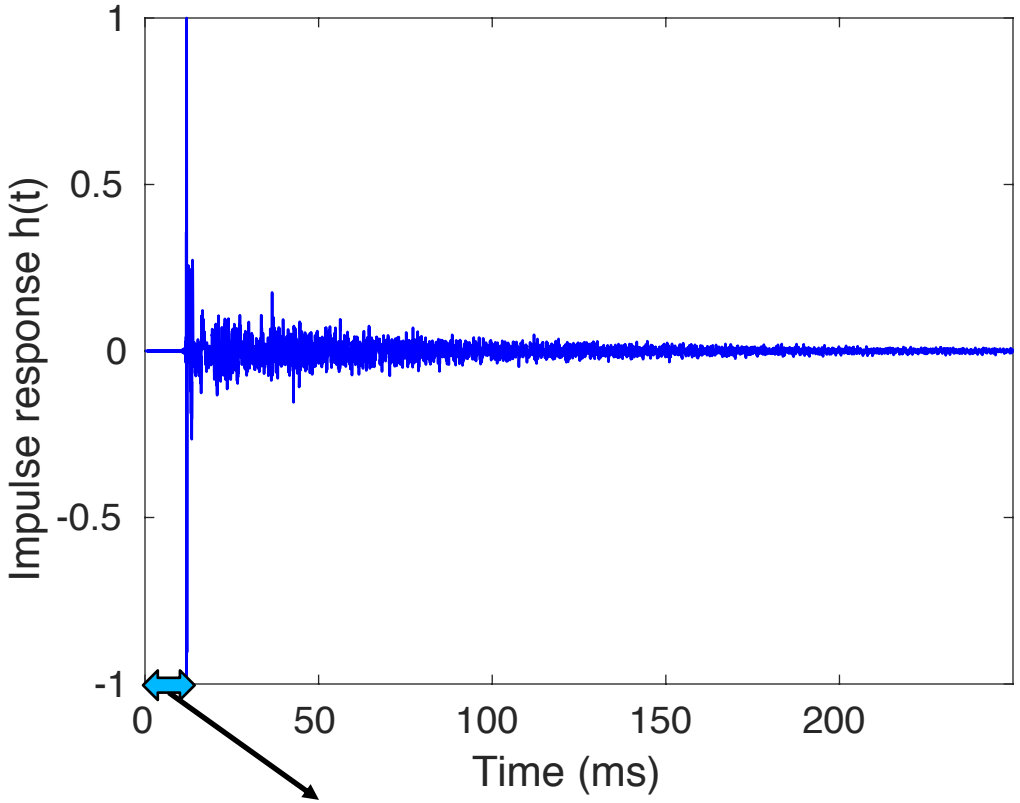
Taking into account the multiple paths due to sound reflections on walls and surfaces, more realistic models are:

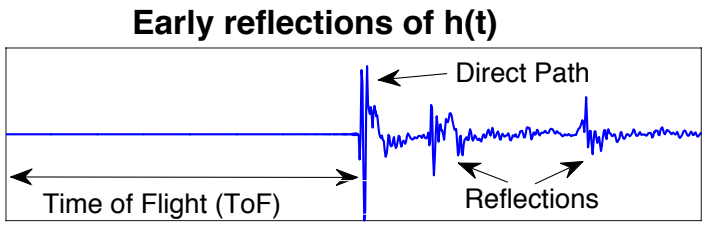
$y_i(t) = x(t) * h_i(\mathbf{s}, t) + n_i(t)$ $h_i(\mathbf{s}, t)$= room impulse response

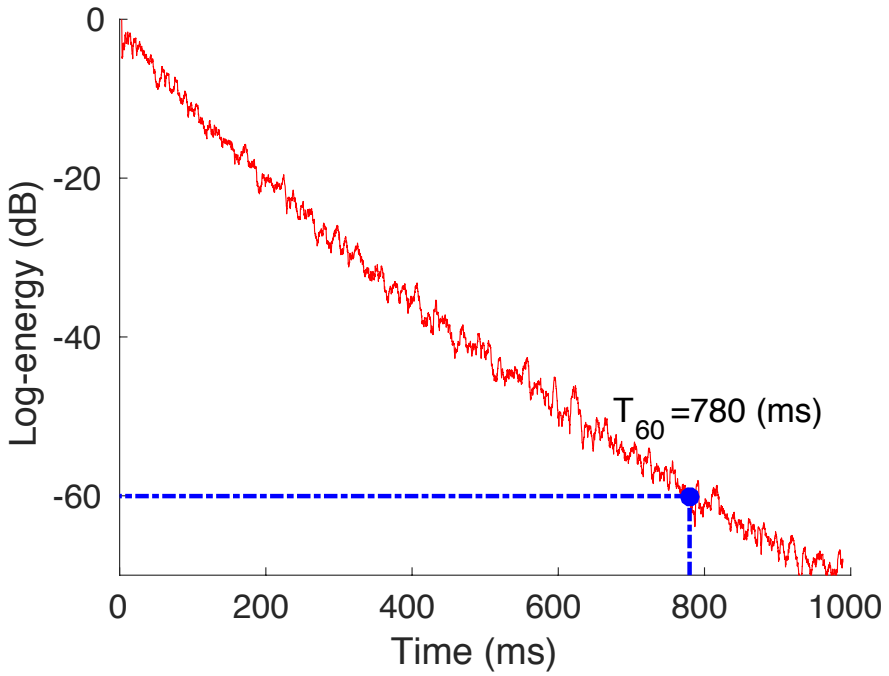
$y_i(t) = \frac{1}{r_i} x(t - T_i) + x(t) * q_i(\mathbf{s}, t) + n_i(t)$ $n_i(t)$ = uncorrelated additive noise

$q_i(\mathbf{s}, t)$ = a room impulse response excluding the direct-path wavefront

Impulse responses, T60 and Direct-to-Reverberant Ratio (DRR)







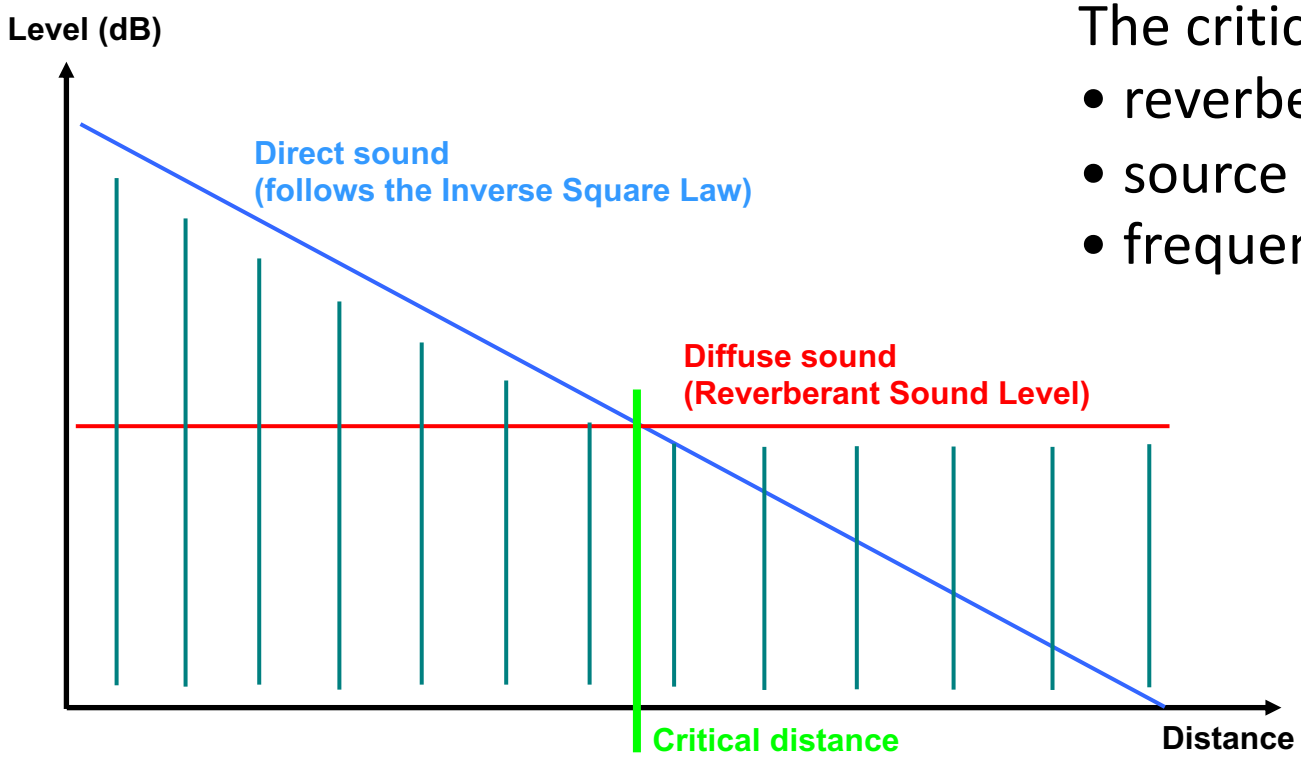
$$DRR = \frac{\int \left| h_m^d(t) \right|^2 dt}{\int \left| h_m^r(t) \right|^2 dt}$$

To derive T60 and DRR estimates from an IR, see for instance: **IR_STATS** (MATLAB code) by Christopher Hummersone, and the referenced paper (Zahorik P. [JASA 2002]).

Critical distance

The ***critical distance*** : distance from the sound source at which the direct and reflected sound intensities are equal.

Beyond the radius of critical distance, Signal to Reverberation Ratio (SRR), generally expressed in dB, becomes negative (except for sound onset)

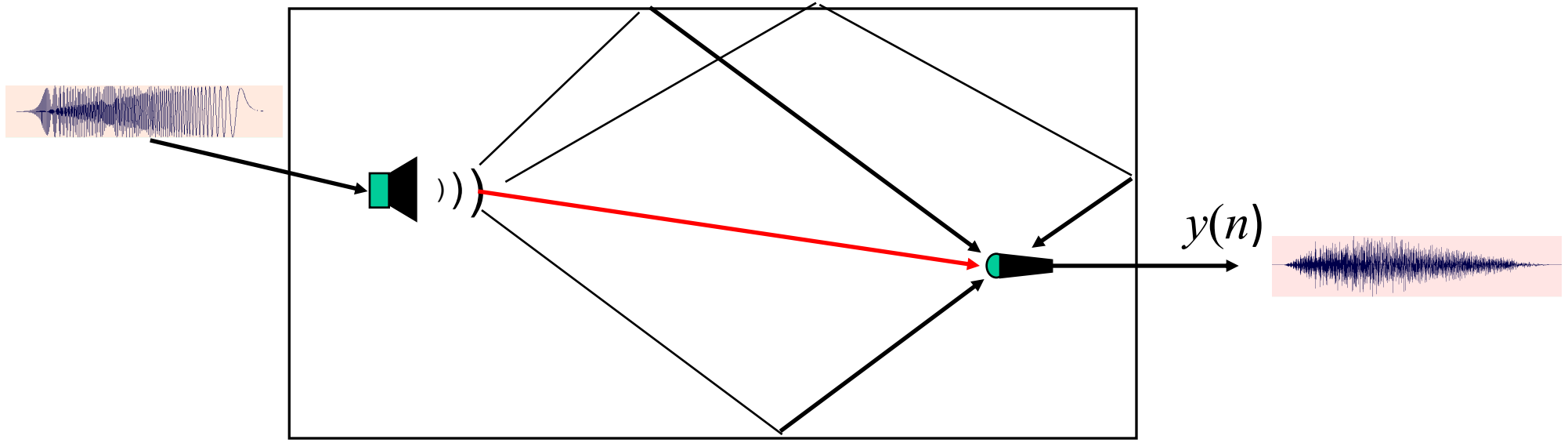


The critical distance depends on:

- reverberation time
- source radiation pattern and orientation
- frequency

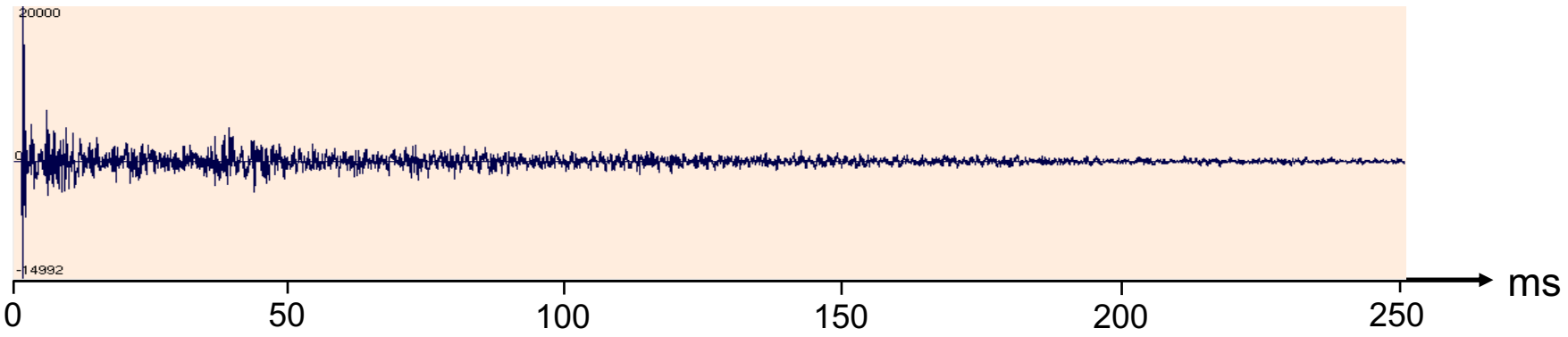
Measurement of an acoustic impulse response

chirp-like sequence $p(n)$



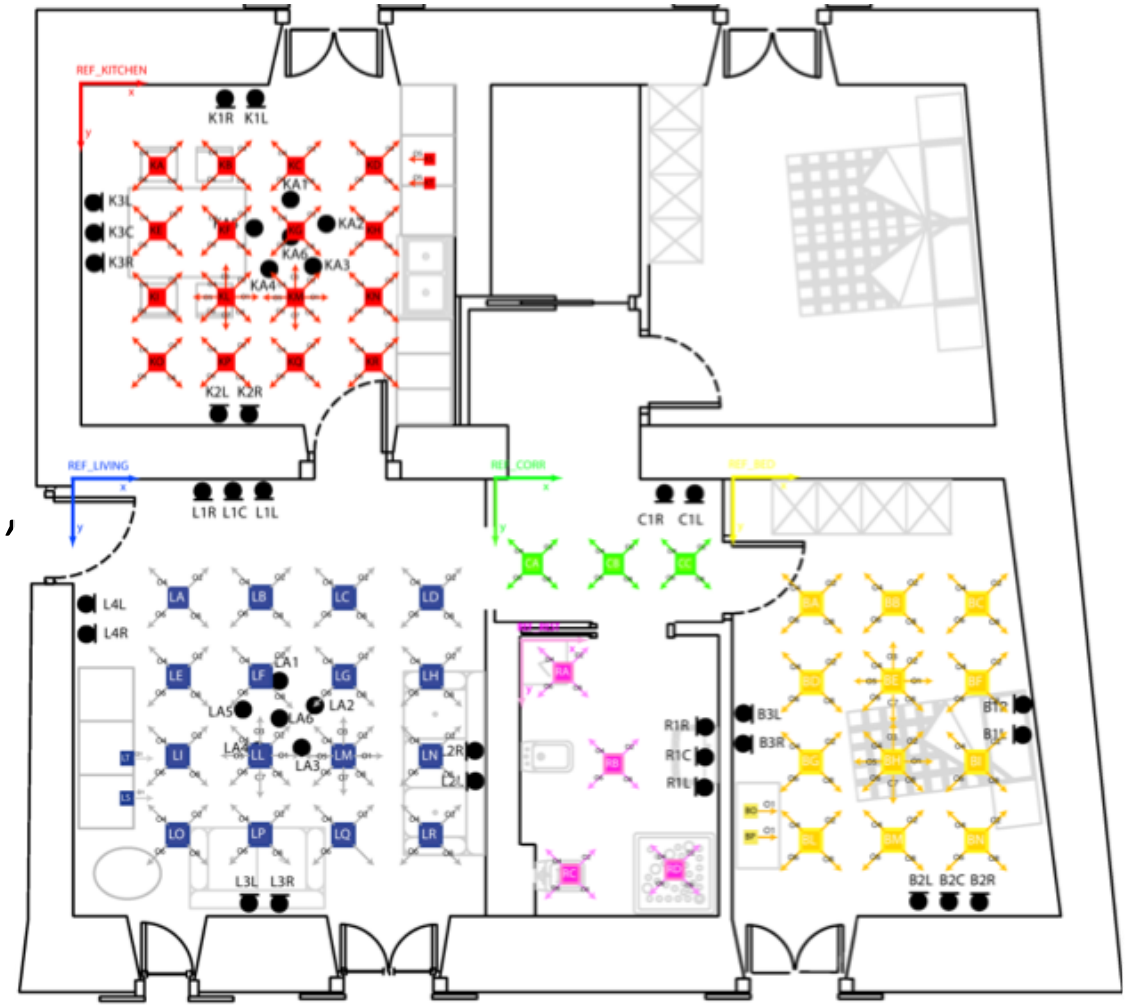
$$h(n)=\text{crosscorr}[p(n),y(n)]$$

Example of IR measured in a real room



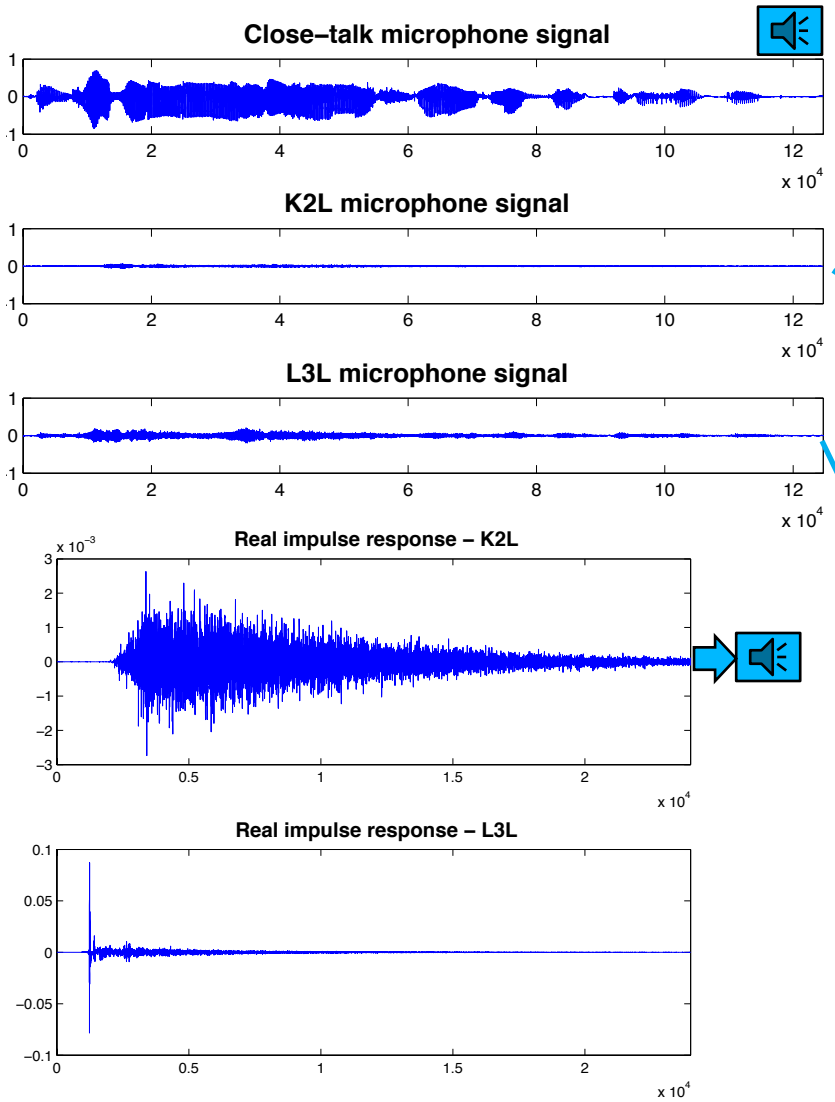
DIRHA apartment set-up and related corpora

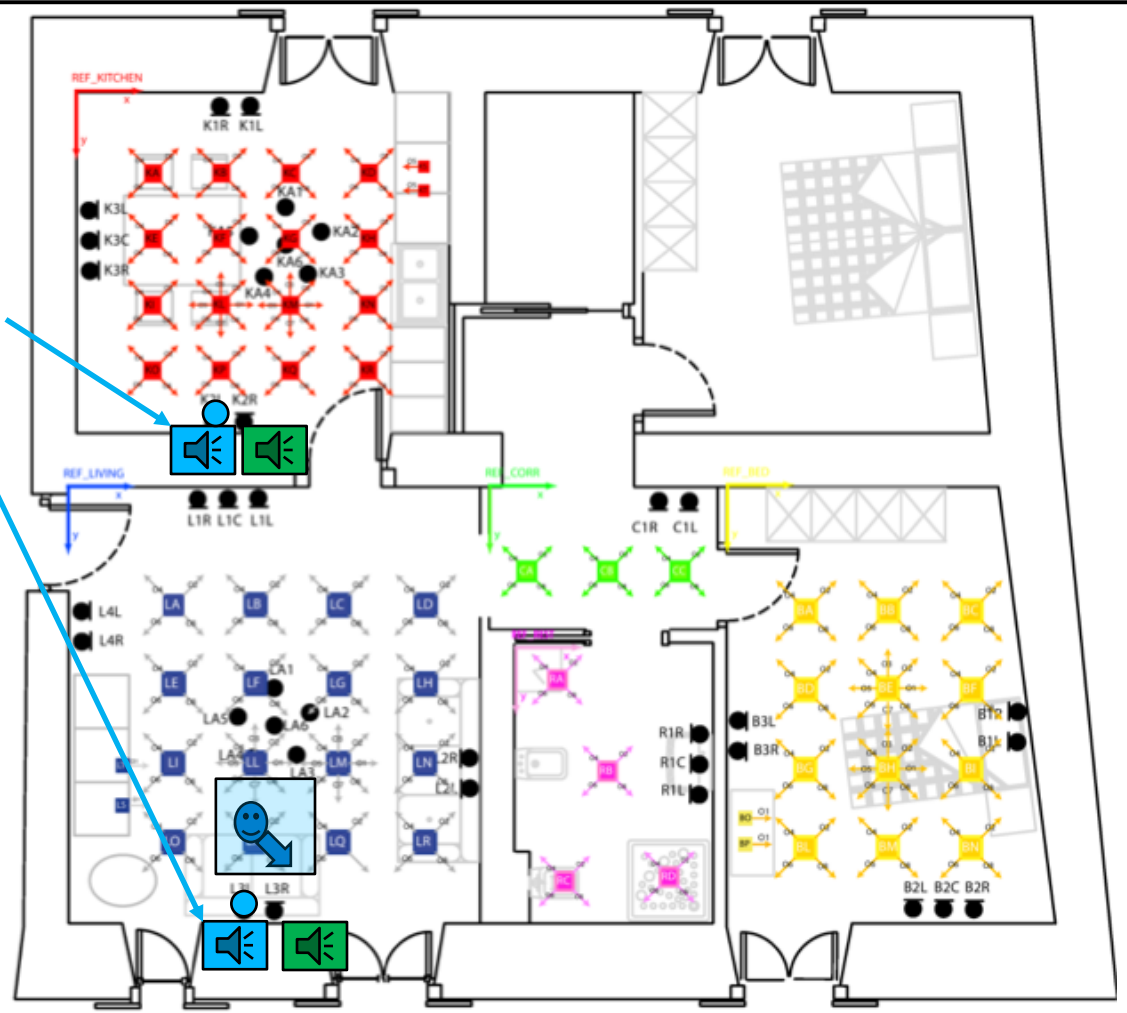
- Microphone arrays and microphone pairs on walls and ceilings
- MEMS digital microphone arrays
- DIRHA simulated and real corpora in Austrian-German, Greek, Italian, Portuguese, US and UK English
- DIRHA US English PhRich, and WSJ0-5k (via LDC)
- About 20.000 real IRs measured in this multi-room environment
- Average room T60 ~ 0.7s



Concerning DIRHA-English, please see http://dirha.fbk.eu and:
M. Ravanelli, et al., "The DIRHA-English corpus and related tasks for distant-speech recognition in domestic environments", ASRU 2015.

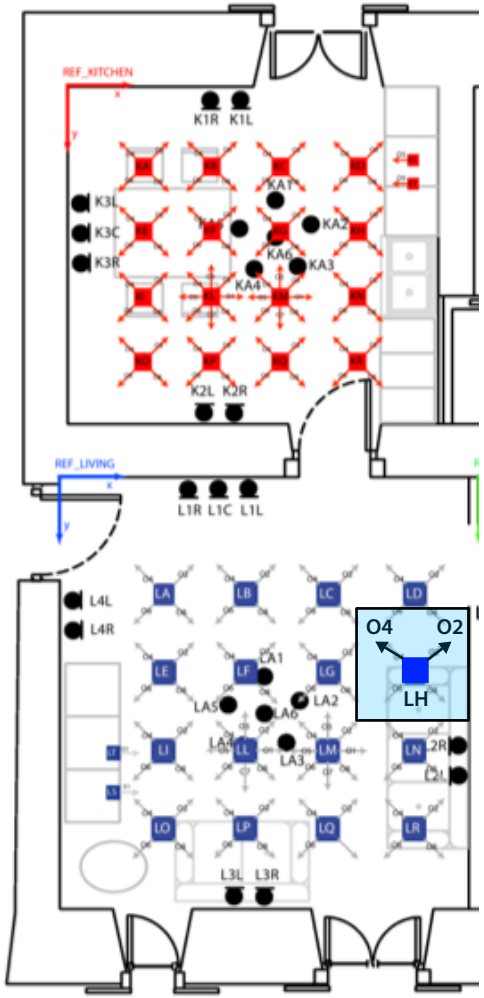
DIRHA real speech corpora: examples in Italian and US English

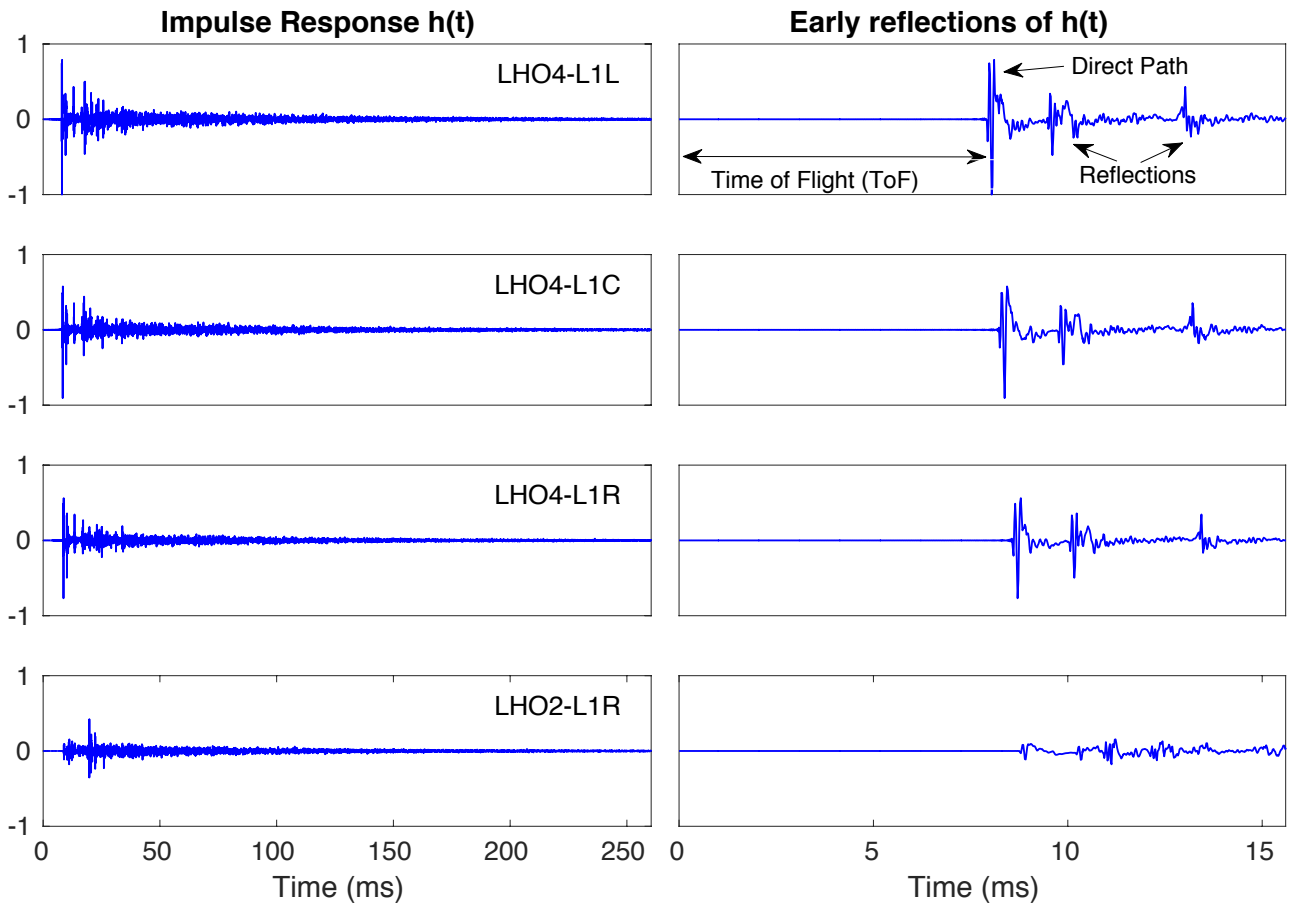




Note the different shape, dynamics, and distribution of peaks observable in real IRs

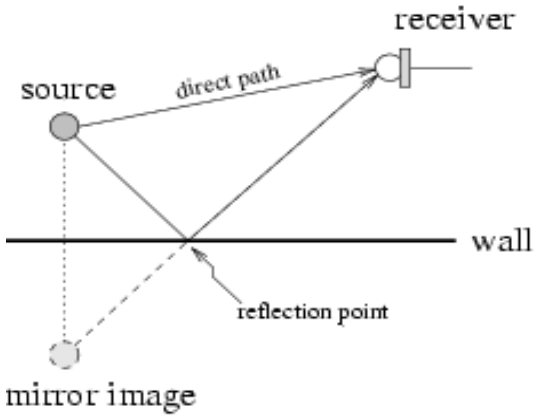
Real IRs: early reflections with different source orientations

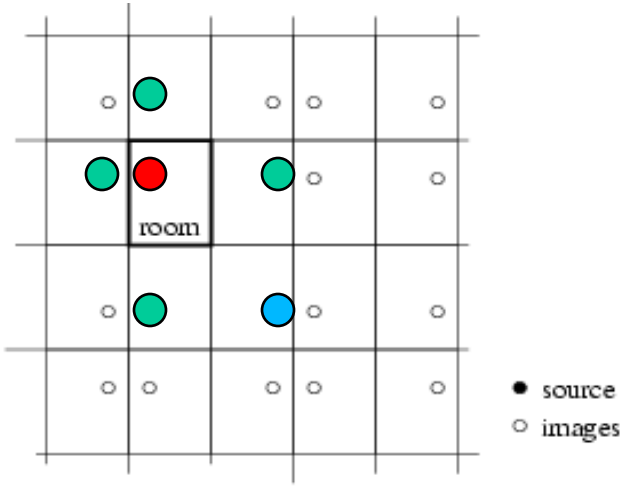


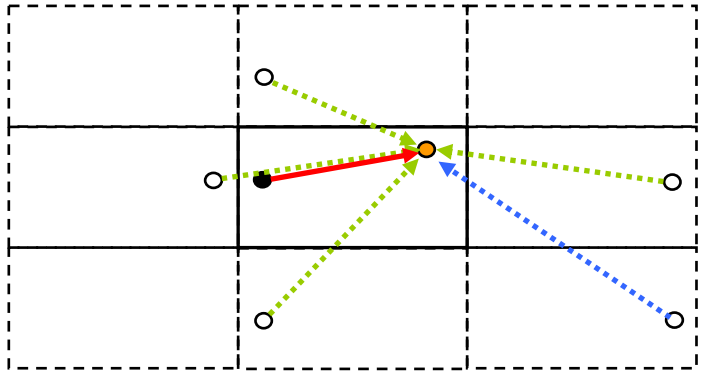


Attenuated direct path and very early reflections correspond to a lower DRR: need of proper modeling it

Sound reflections in an enclosure: the Image method







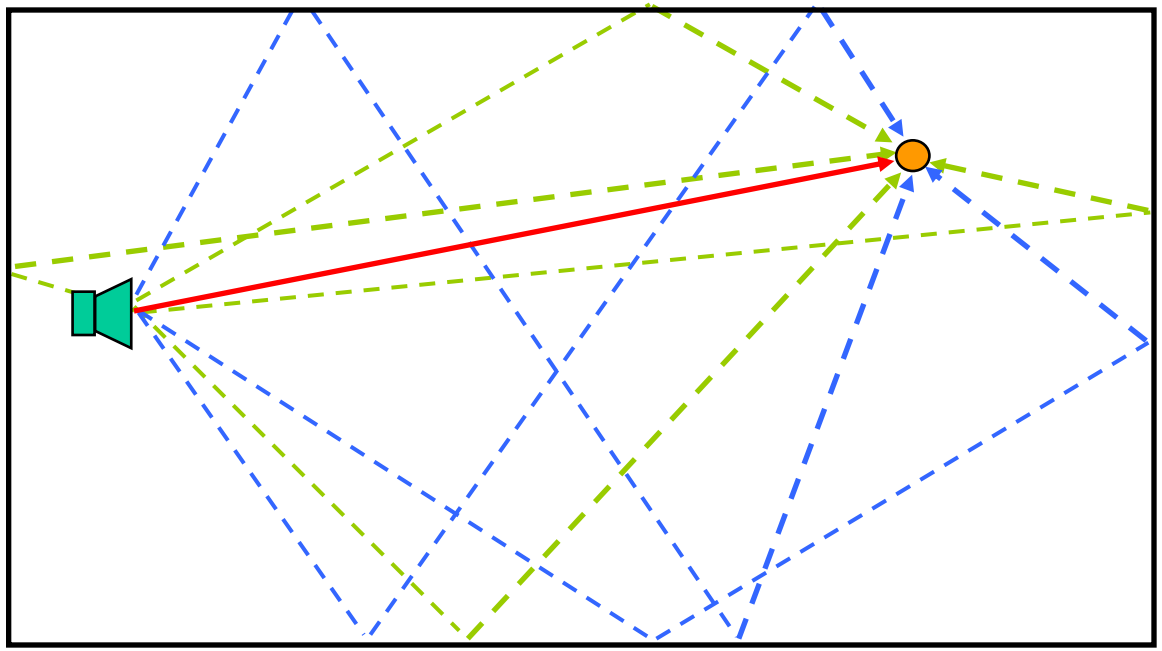
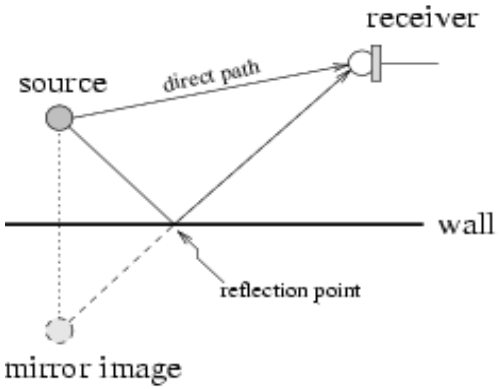
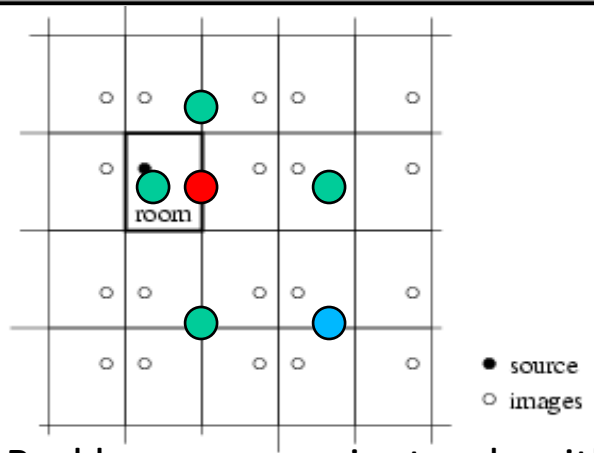


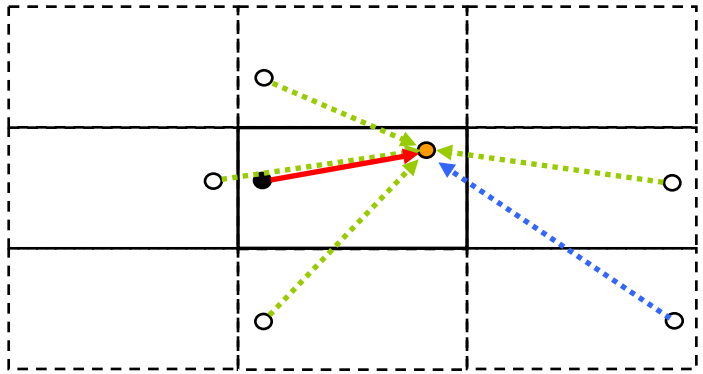
Image method: useful to simulate propagation in a room with given surface reflection characteristics (see [Allen-Berkley'79]). It starts from the wave equation that governs the propagation of sound waves in a lossless fluid and Green's function for a rectangular room with rigid walls.

Based on it, and on clean speech of good quality, realistic artificial data sets can be derived

Simulate reverberant environments: IM-based modeling







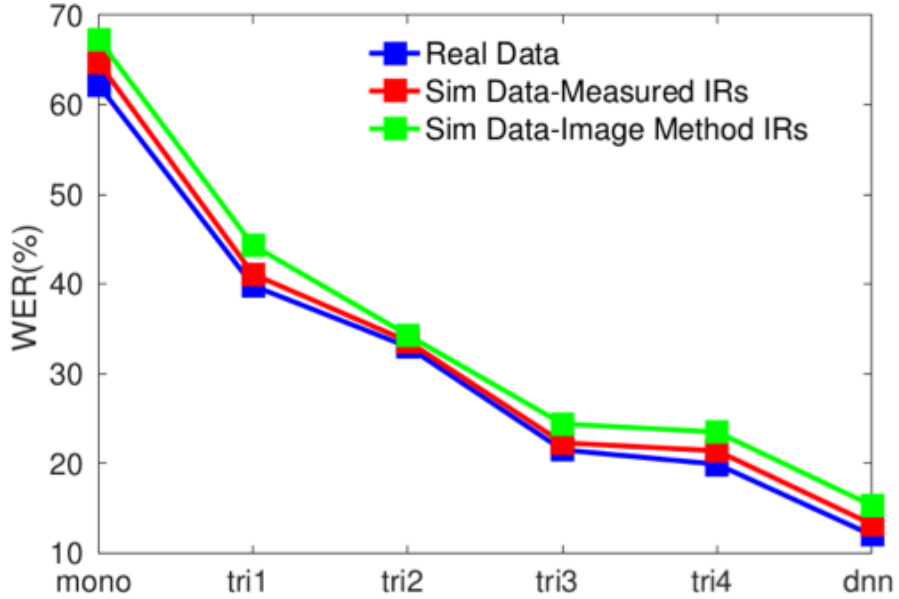
See more detailed aspects in the related lecture lab of this afternoon

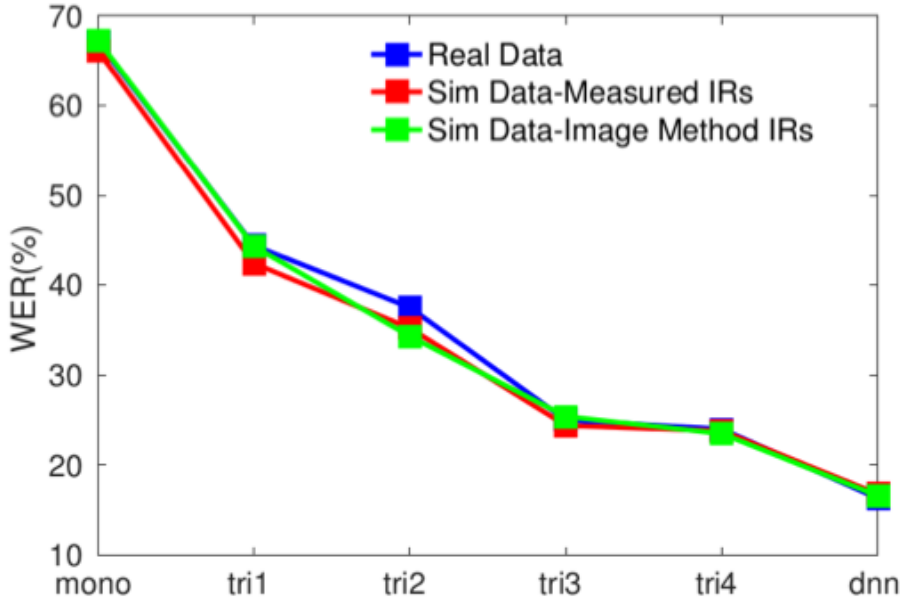
From the original work of Allen-Berkley, many variants, algorithms, and very useful software tools:

- J. Allen and D. Berkley, "*Image method for efficiently simulating small room acoustics*", JASA, 65-4, pp. 943-950, 1979
- E. Lehmann and A. Johansson, "*Prediction of energy decay in room impulse responses simulated with an image-source model* ", JASA, vol. 124, n. 1, pp. 269-277, 2008.
- E. Habets, *"Room impulse response generator*", Technical Report (see https://github.com/ehabets/RIR-Generator/blob/master/rir_generator.pdf).
- S. G. Mc Govern, *"Fast image method for impulse response calculations of box-shaped rooms",* Appl. Acoust., vol. 70, n.1, pp. 182-189, 2009.
- S. Hafezi, *"Room impulse response for directional source (RIRD) generator",(see* http://www.ee.ic.ac.uk/sap/rirdgen). and others, including the one we will see in the afternoon. See also, other techniques in:
- L. Savioja and U.P. Svensson, Overview of geometrical room acoustic modeling techniques, JASA, 138, 708, 2015
- V. Valimaki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty years of artificial reverberation," IEEE Trans. on Audio, Speech and Language Process., vol. 20, no. 5, pp. 1421–1448, 2012.

Kaldi-based distant speech recognition experiments

- Test on DIRHA-English WSJ0-5k distant-speech material
- Training based on either *real IR* (left) or *directional-IM* (right), in both cases filtering original WSJ0 training signals (see our Interspeech 2016)





- Very similar performance trend with test material of different nature (matching confirmed with microphone array and D&S beamforming)
- With DNN:

IRs used in training	IM-omni	IM-directive	Measured
WER (%)	13.0	12.5	11.9

Multi-microphone data sets and tasks – Part I

JSALT 2019 School – June 19th, 2019

Maurizio Omologo, Fondazione Bruno Kessler (FBK), Trento, Italy

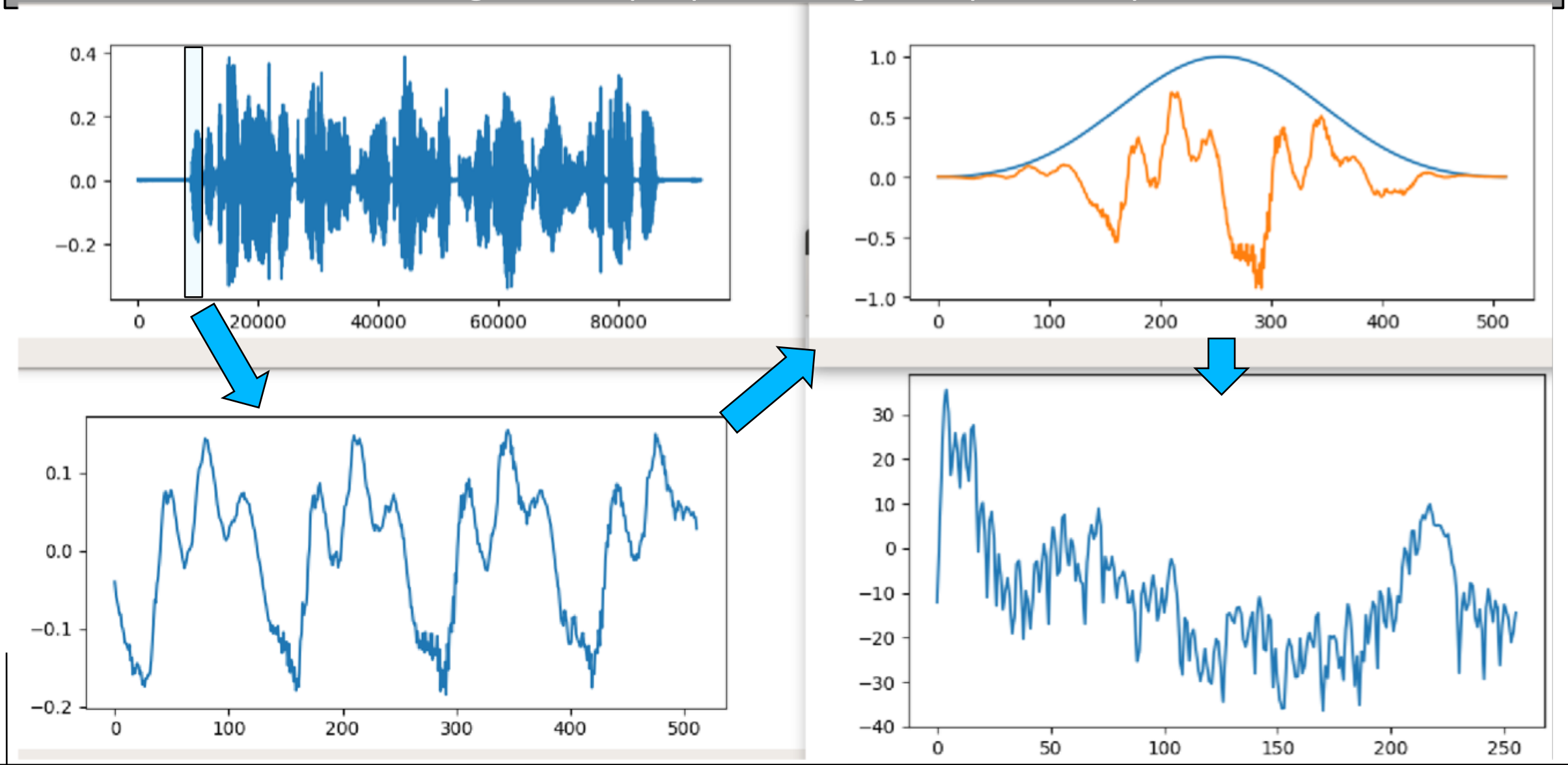
Agenda of the day

9:00 – 10:30 AM Multi-microphone signal processing for distant-speech recognition (DSR) - Maurizio Omologo
10:30 – 10:50 AM Break
10:50 AM – 12:10 PM Cooperative and self-supervised neural frameworks for DSR – Part I - Mirco Ravanelli
12:10 – 1:00 PM Lunch Break
1:00 – 2:00 PM Cooperative and self-supervised neural frameworks for DSR – Part II - Mirco Ravanelli
1.30 – 2.00 PM PyTorch Kaldi - Mirco Ravanelli
2:00 – 3:00 PM **Multi-microphone data-sets and tasks – Part I** - Maurizio Omologo
3:00 – 3:30 PM Coffee Break
3:30 – 4:30 PM Multi-microphone data-sets and tasks – Part II - Maurizio Omologo
4.30 - 5.00 PM Questions and conclusions

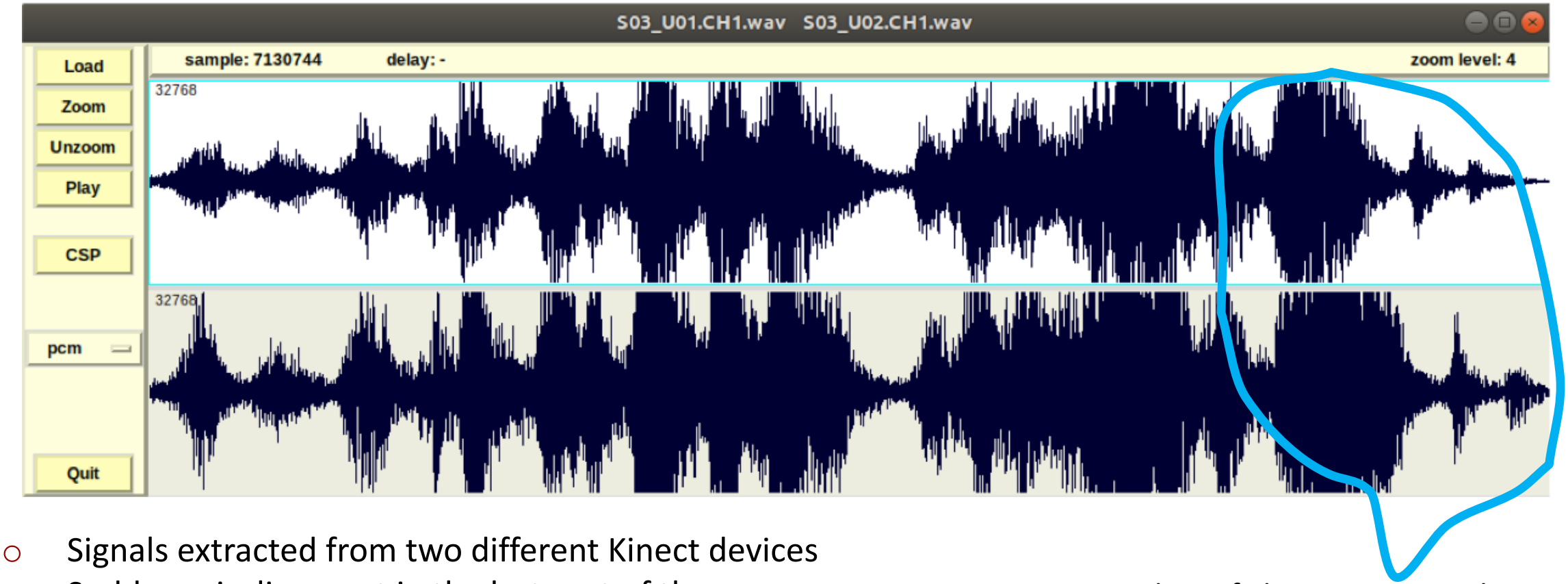
Outline of this first part of lab-lecture

- Editing and simple processing of a speech sequence (examples with Audacity, Praat, and in Python)
- Examples of multi-microphone signals extracted from CHiME5 corpus
- Application of the image-method
- Examples of toy CHiME5-like artificial conversations

Editing and simple processing of a speech sequence



Sample loss: example extracted from CHiME5 corpus

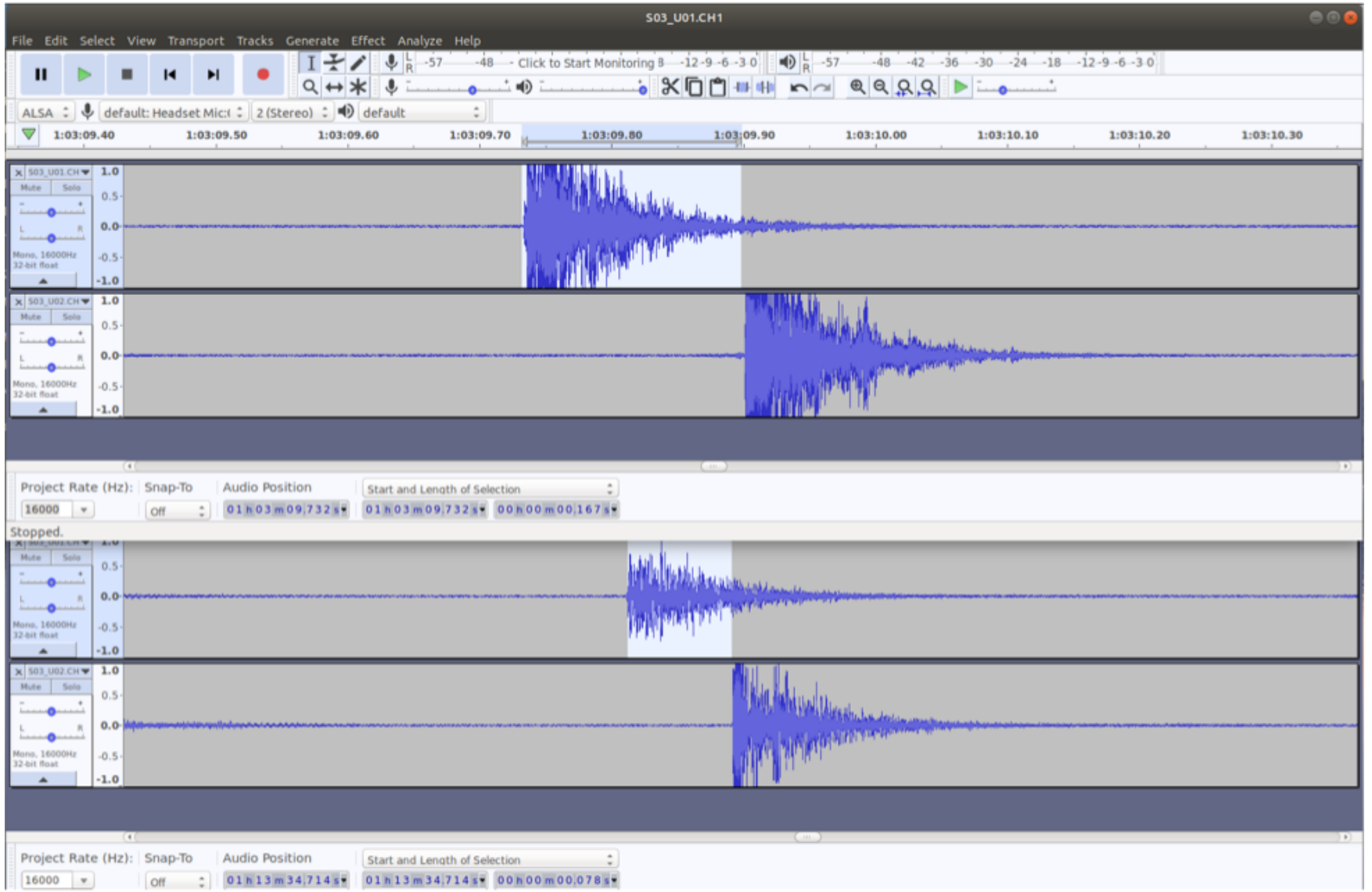


- Signals extracted from two different Kinect devices
- Sudden misalignment in the last part of the sequence
- Crucial aspect for array processing (e.g., beamforming)
- A target during the workshop: exploit redundancy among all signals, to detect sample loss, estimate how many samples were lost, and realign. Need to obtain a very accurate loss estimate

loss of about 1000 samples

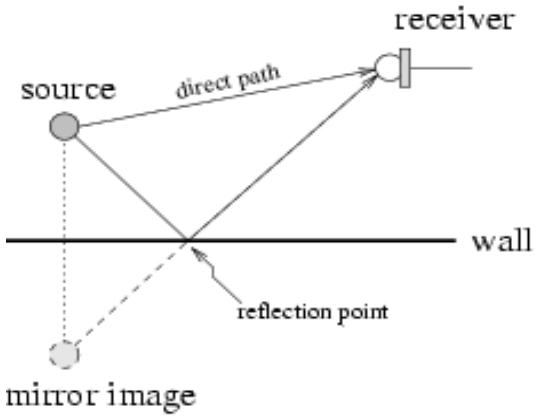
Sample loss detection: inspection via Audacity

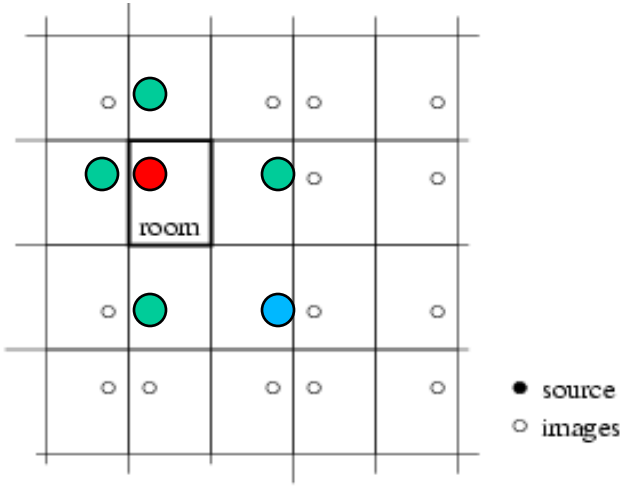
- Estimation of loss of samples for a segment of a CHiME5 session (i.e. S03)
- The sequences in the upper part are characterized by a shift of about 167 ms, which becomes about 78 ms in the lower part (i.e. after about 10 minutes)

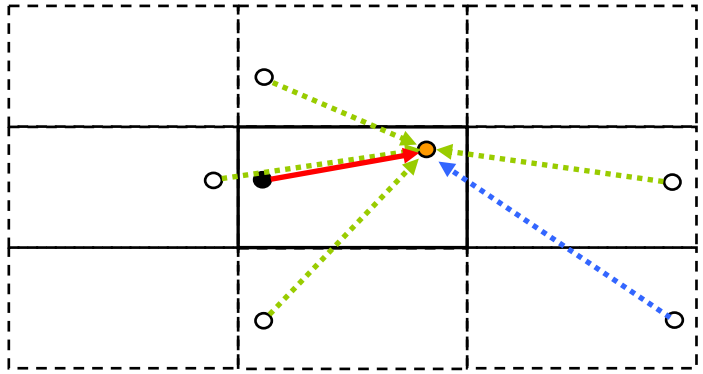


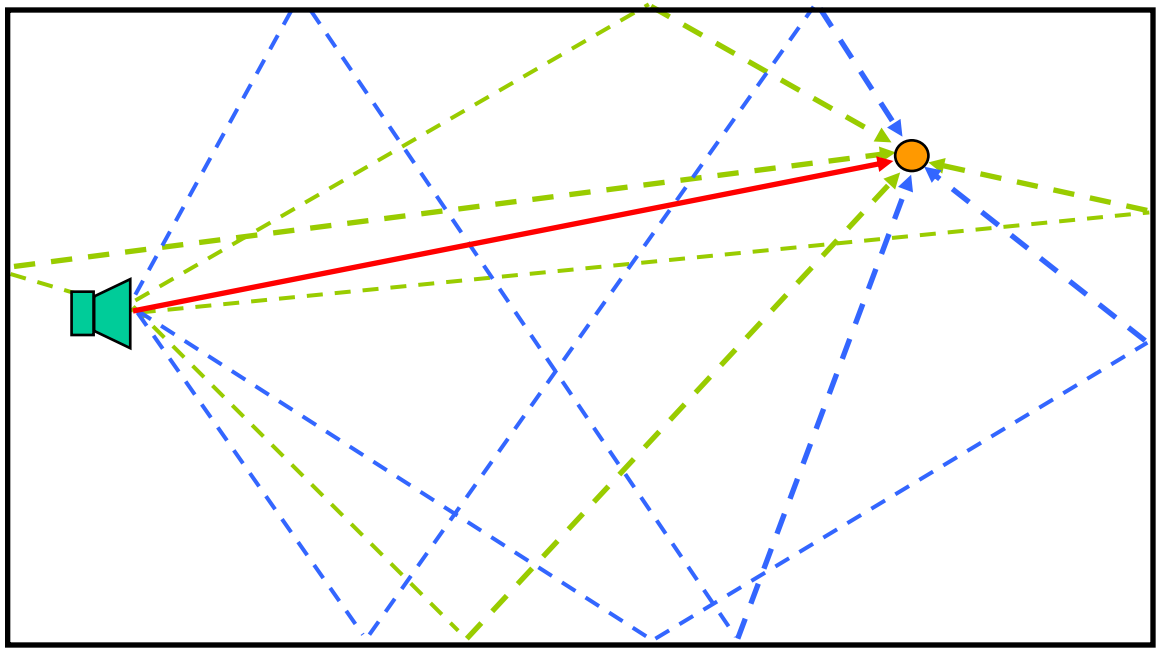
Maurizio Omologo - omologo@fbk.eu

Sound reflections in an enclosure: the Image method

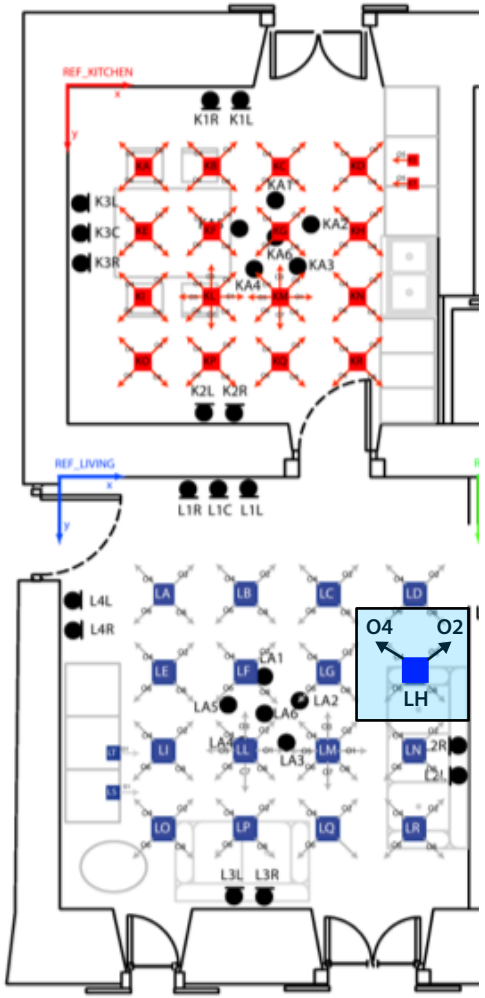


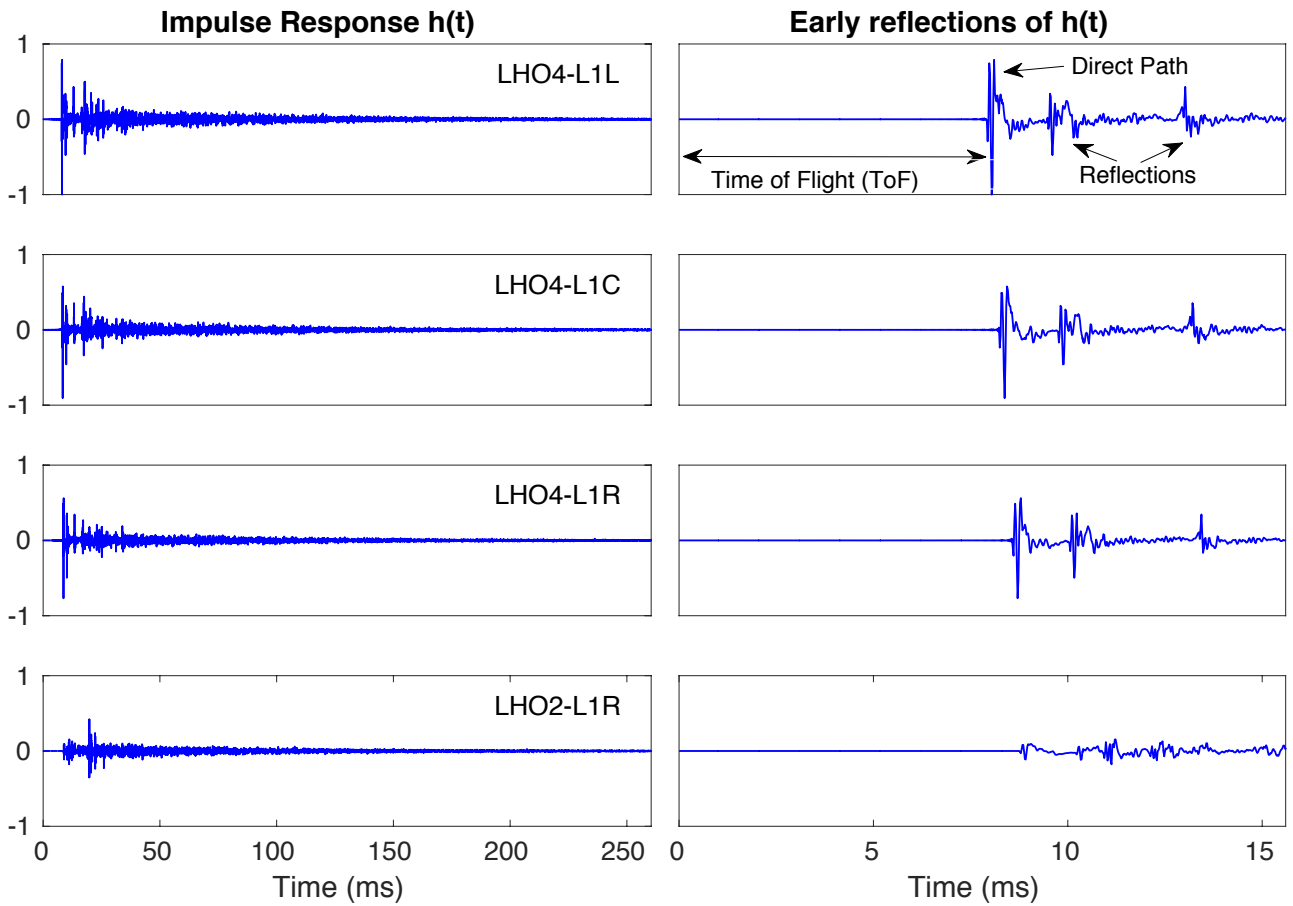






Real IRs: early reflections with different source orientations

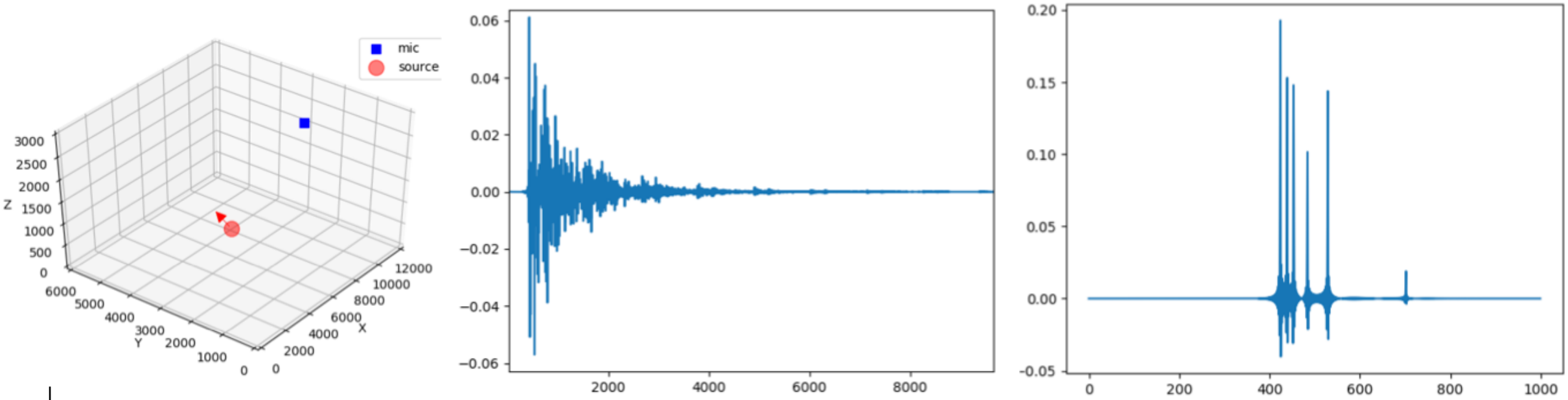




Attenuated direct path and very early reflections correspond to a lower DRR: need of proper modeling it

Application of the image method

- From order 1 to a higher order
- From most traditional version in the time-domain to frequency-domain
- From omni-directional source to directional source
- Changes in the source orientation vs early reflections
- From low to high T60s

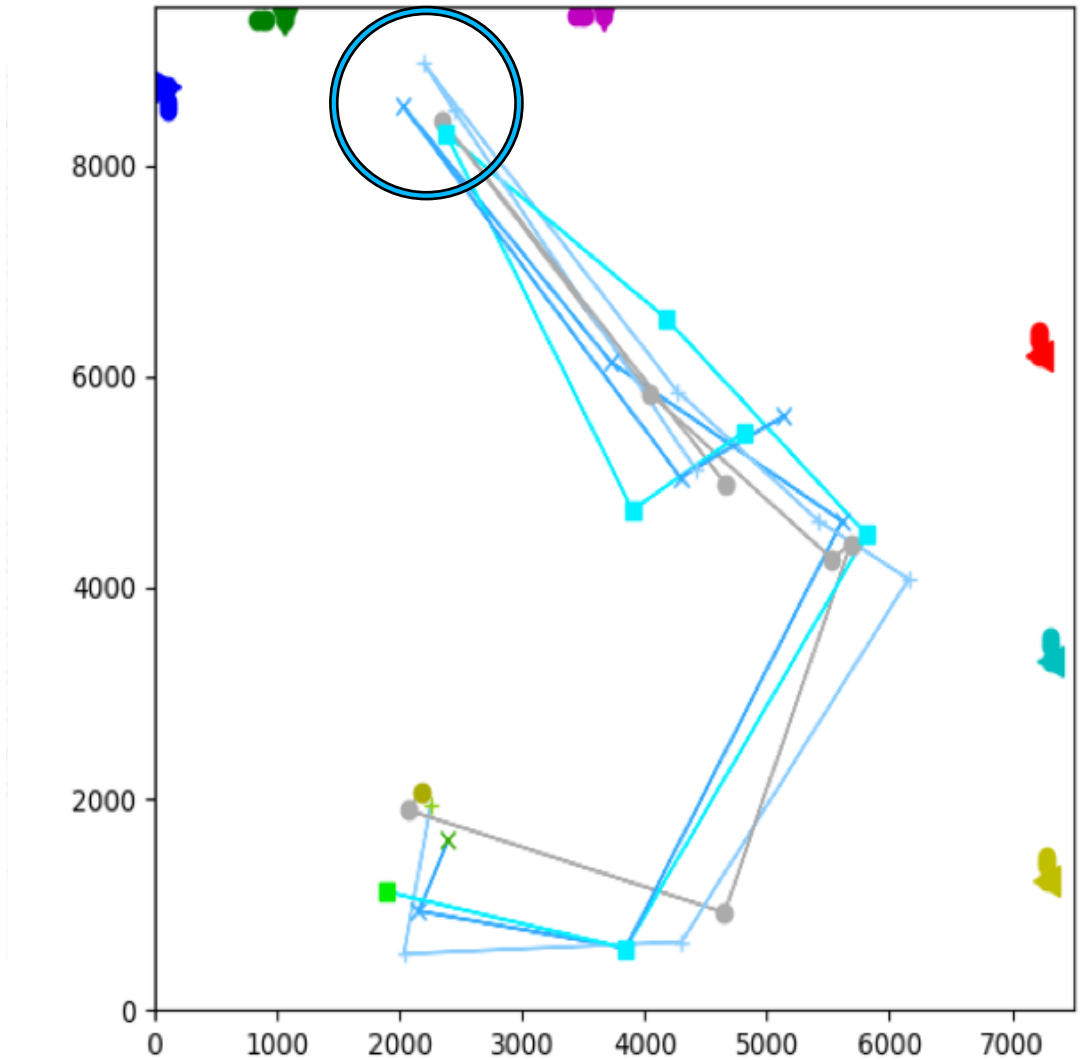


Example of impulse responses computed with order 200 (center), with order 1 (right), for the geometry shown in the picture at left.

Multi-channel artificial conversations based on Librispeech

- Corpus conceived for initial studies and toy experiments
- Simulated scenes similar to those of CHiME5:
  - 6x4 microphones (same Kinect geometry), 4 speakers, 16 kHz sampling frequency
- Variable room size, and reverberation characteristics
- Average duration 5-6 minutes
- Speakers change location after every pause
- 3D simulation of environmental acoustics and diffused noise
- Directive speaker sources
- Overlapped speech
- Segmentation boundaries at phone level for each speaker activity (based on the output of the Montreal Forced Aligner (see https://montreal-forced-aligner.readthedocs.io)
- Possible simulation of clock drift between devices and of sample loss
- Targeted version based on fully using Librispeech dev and test portions
- Available reverberated (and noisy) speech from Librispeech – train100 for training
- xml-files including scene descriptions

Geometry of a scene and details of the related xml




```
      <X> 2345 </X>
      <Y> 8428 </Y>
      <Z> 1500 </Z>
      <Or_Az> 302 </Or_Az>
      <Or_El> 89 </Or_El>
    </SourceLabXYZ>
    <SourceLabXYZ>
      <Index> 23 </Index>
      <Sp_Index> 0 </Sp_Index>
      <X> 2382 </X>
      <Y> 8295 </Y>
      <Z> 1500 </Z>
      <Or_Az> 50 </Or_Az>
      <Or_El> 63 </Or_El>
    </SourceLabXYZ>
    <SourceLabXYZ>
      <Index> 24 </Index>
      <Sp_Index> 1 </Sp_Index>
      <X> 2025 </X>
      <Y> 8565 </Y>
      <Z> 1500 </Z>
      <Or_Az> 205 </Or_Az>
      <Or_El> 97 </Or_El>
```

Multi-microphone data sets and tasks – Part II

JSALT 2019 School – June 19th, 2019

Maurizio Omologo, Fondazione Bruno Kessler (FBK), Trento, Italy

Agenda of the day

9:00 – 10:30 AM Multi-microphone signal processing for distant-speech recognition (DSR) - Maurizio Omologo
10:30 – 10:50 AM Break
10:50 AM – 12:10 PM Cooperative and self-supervised neural frameworks for DSR – Part I - Mirco Ravanelli
12:10 – 1:00 PM Lunch Break
1:00 – 2:00 PM Cooperative and self-supervised neural frameworks for DSR – Part II - Mirco Ravanelli
1.30 – 2.00 PM PyTorch Kaldi - Mirco Ravanelli
2:00 – 3:00 PM Multi-microphone data-sets and tasks – Part I - Maurizio Omologo
3:00 – 3:30 PM Coffee Break
3:30 – 4:30 PM **Multi-microphone data-sets and tasks – Part II** - Maurizio Omologo
4.30 - 5.00 PM Questions and conclusions

Outline of this second part of lab-lecture

- Speaker location and tracking: from TDOA estimation to acoustic maps
  - GCC-PHAT analysis*
  - Microphone array polar patterns*
  - GCF acoustic maps
  - Examples based on the artificial CHiME5-like conversation corpus
- Possible use of GCC-PHAT related features for speech activity detection
- Ad-hoc microphone arrays: the clock-drift problem

* I will use two tools that were developed by Piergiorgio Svaizer (FBK, Trento, Italy)

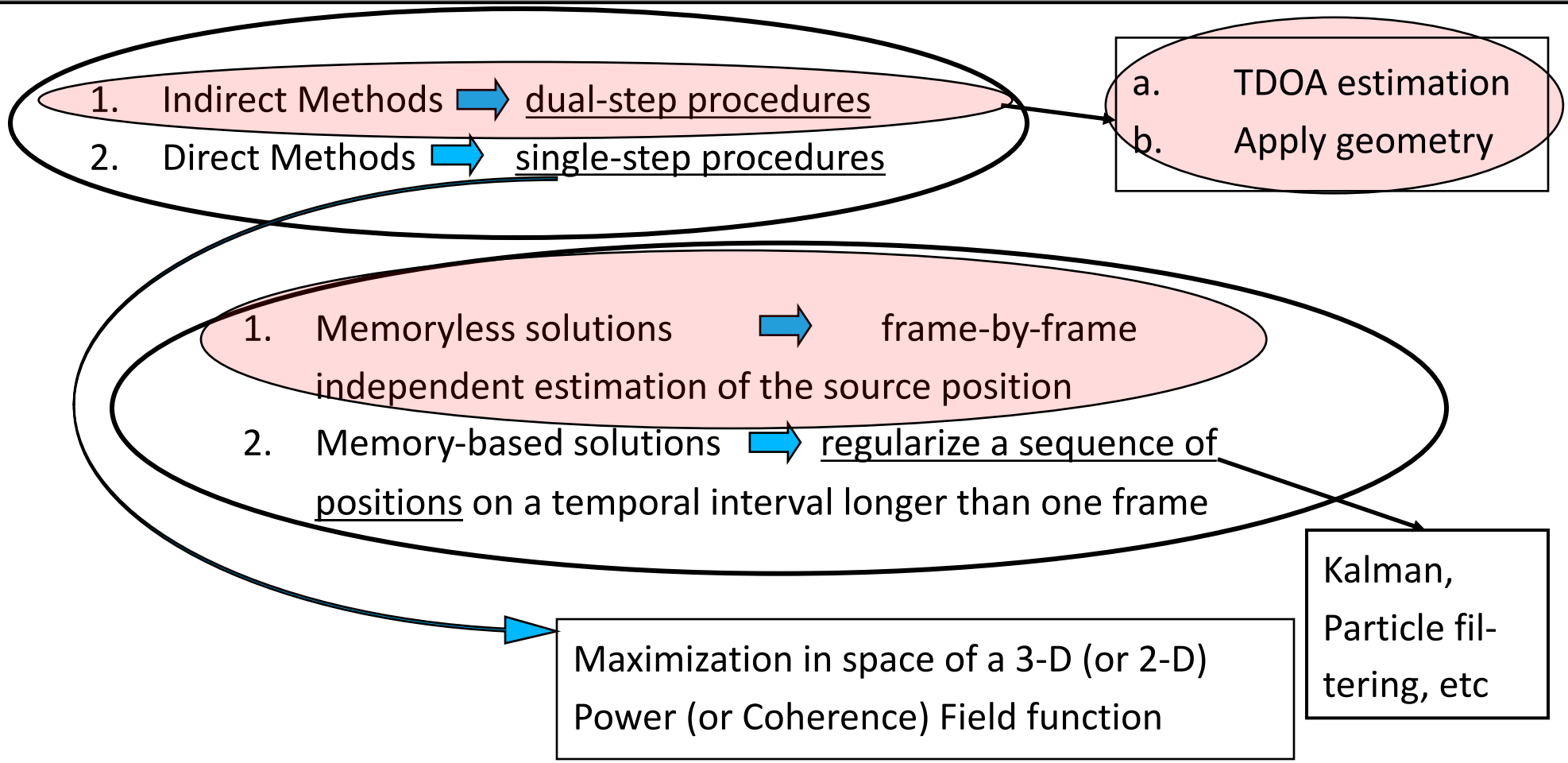
Speaker location for acoustic scene analysis

- Acoustic scene analysis with a microphone network includes many different possible sub-tasks, including sound source location
  - from single to multiple sources
  - from stationary to moving sources (i.e., tracking)
  - from omnidirectional to directional sources
- Locating a sound source makes sense only:
  - if its size is sufficiently small to be approximated to a point source
  - for temporal segments when it is active (see acoustic event detection and, in particular, the related task of speech activity detection)
- Sound source location, or extraction of location cues, is tightly related to tasks such as:
  - beamforming (e.g., delay-and-sum)
  - source separation, speech enhancement
  - self-calibration (possibly to deduce unknown microphone positions)
  - combination with visual input in a multi-modal analysis framework
  - estimation of the source orientation
  - speaker identification

Speaker Location (SLOC) and Tracking

- ✓ *Most of the research activities since 1990*
- ✓ *Early technologies inspired by binaural sound source localization (mostly based on interaural time difference)*
- ✓ *The most critical issue: derive a Time Difference of Arrival with high accuracy from a microphone pair input*
- Other major issues:
  - Microphone array **Geometry**
  - **Quantity** and **Quality** of the microphones
  - Characteristics of **Environmental Noise** and **Reverberation**
  - Number of **Active Sources** and related spectral contents
  - **Head Orientation** (or radiation pattern of a generic source)
  - Combine Speaker Location, with **Speaker ID**, and **Acoustic Event Detection**
  - System **Promptness** (even with short events, overlapping each other)

Acoustic source location: basic approaches and techniques

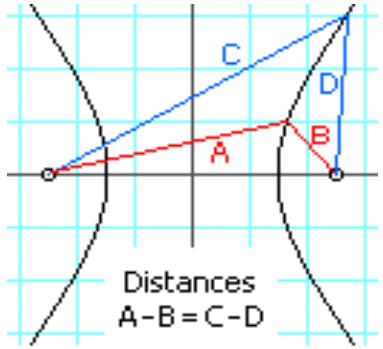


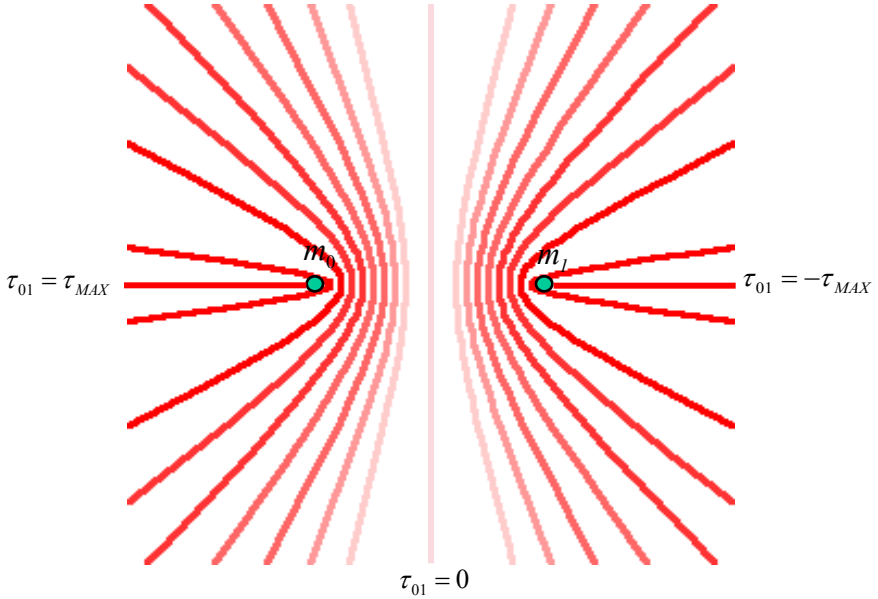
Microphone pair: TDOA-based location in 2D

The locus of potential source positions **s**=[s_x,s_y] associated to a delay τ_{01} in a 2D space, satisfies:

$$\tau_{01} = \frac{\left|\mathbf{s}-\mathbf{m}_1\right| - \left|\mathbf{s}-\mathbf{m}_0\right|}{c}$$

Range difference = $c\tau_{01}$



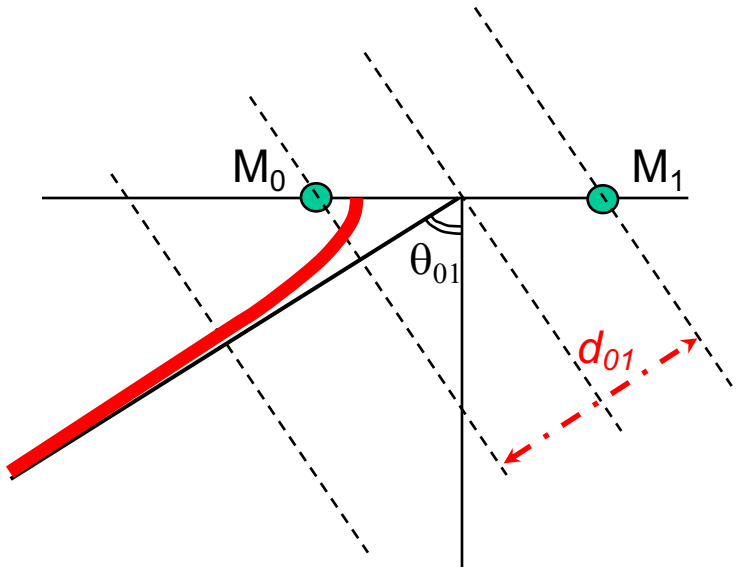


It is a family of hyperbolas parameterized on τ_{01} with foci on the two microphones

For positive delays the source is on the left curves, for negatives delays on the right curves.

In a far-field situation the hyperbolas can by approximated by their asymptotes.

Trivial two-step solution based on two microphone pairs



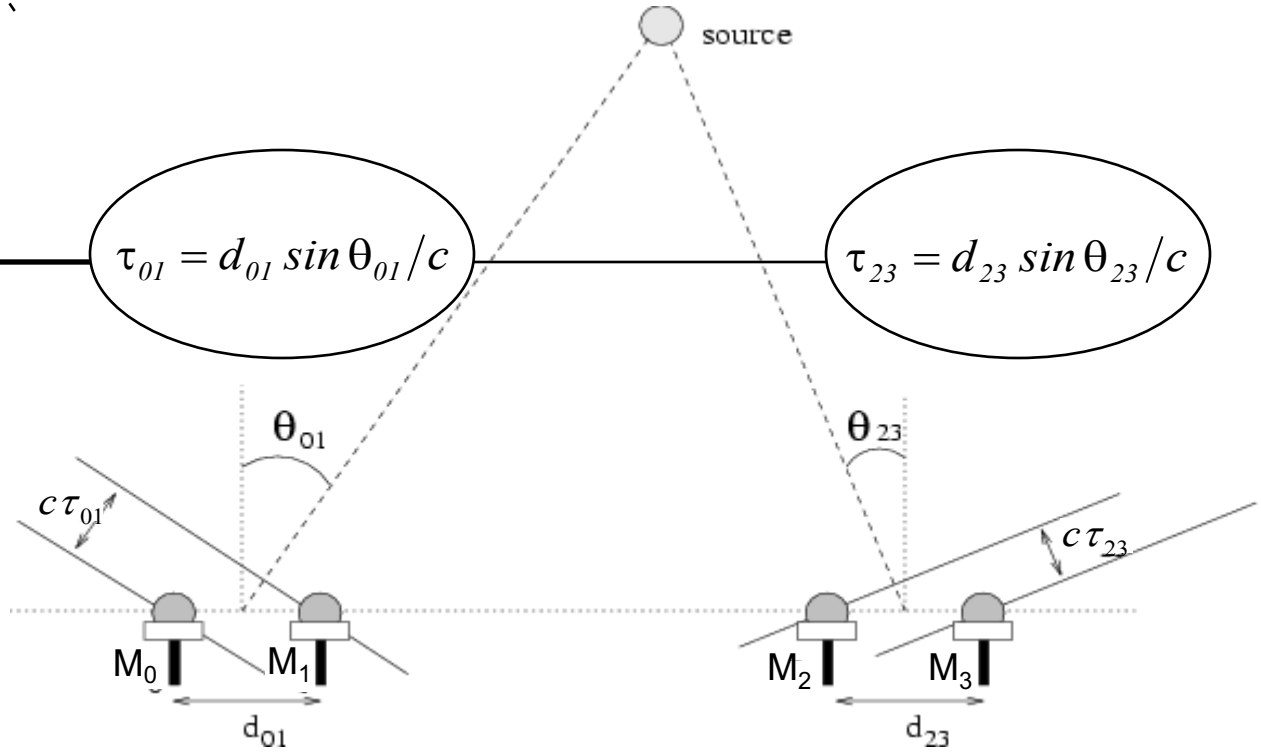
$$\theta_{01} = arcsin\left(\frac{c\,\tau_{01}}{|m_1 - m_0|}\right)$$

$$d_{01} = c\tau_{01}$$

a. Estimate the two delays
b. Cross the resulting directions

$\tau_{01} = d_{01}\sin\theta_{01}/c$

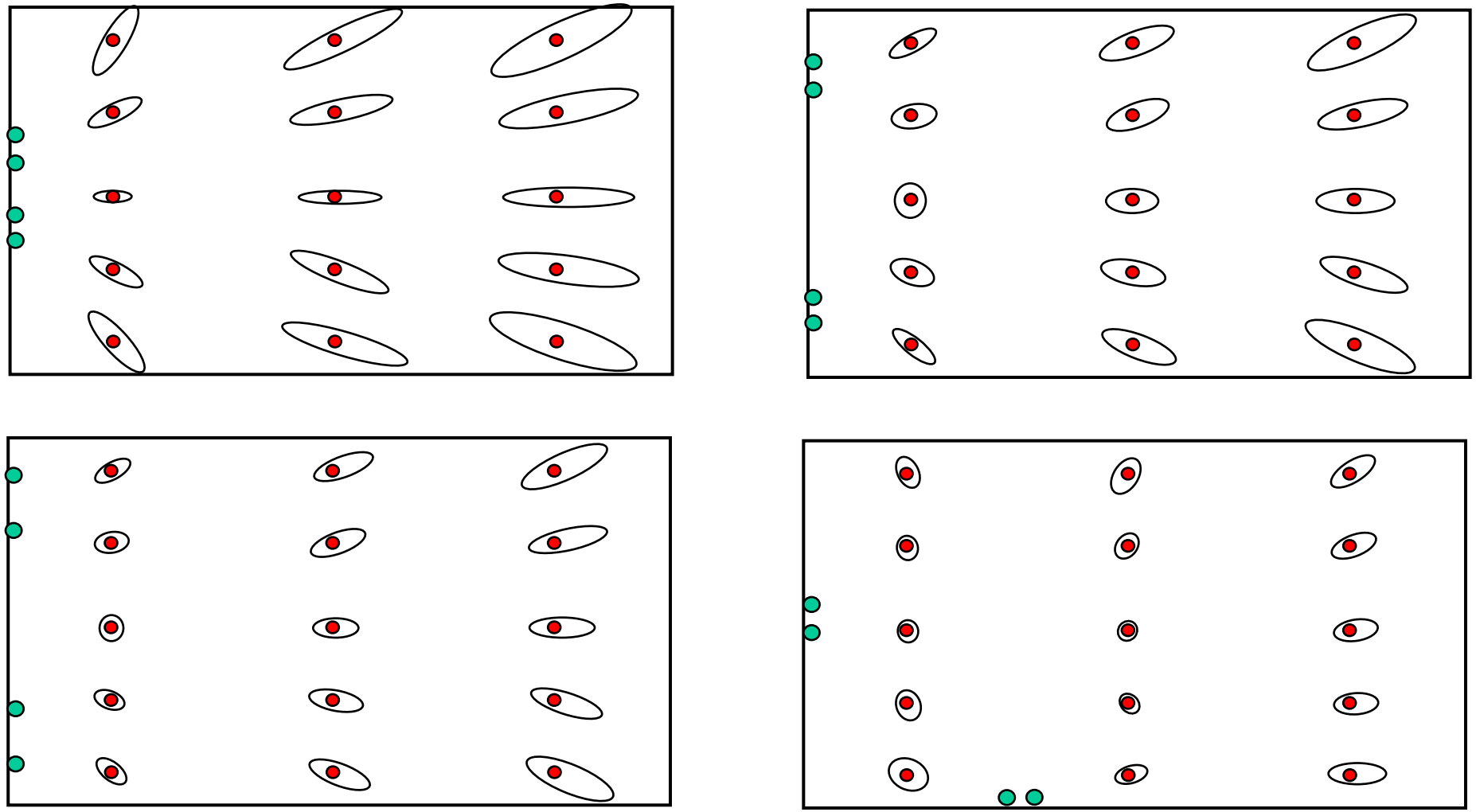
$\tau_{23} = d_{23}\sin\theta_{23}/c$



TDOA error statistics vs location accuracy

Changing the array geometry vs potential location accuracy

Variance of location estimate changes as a function of the microphone placement



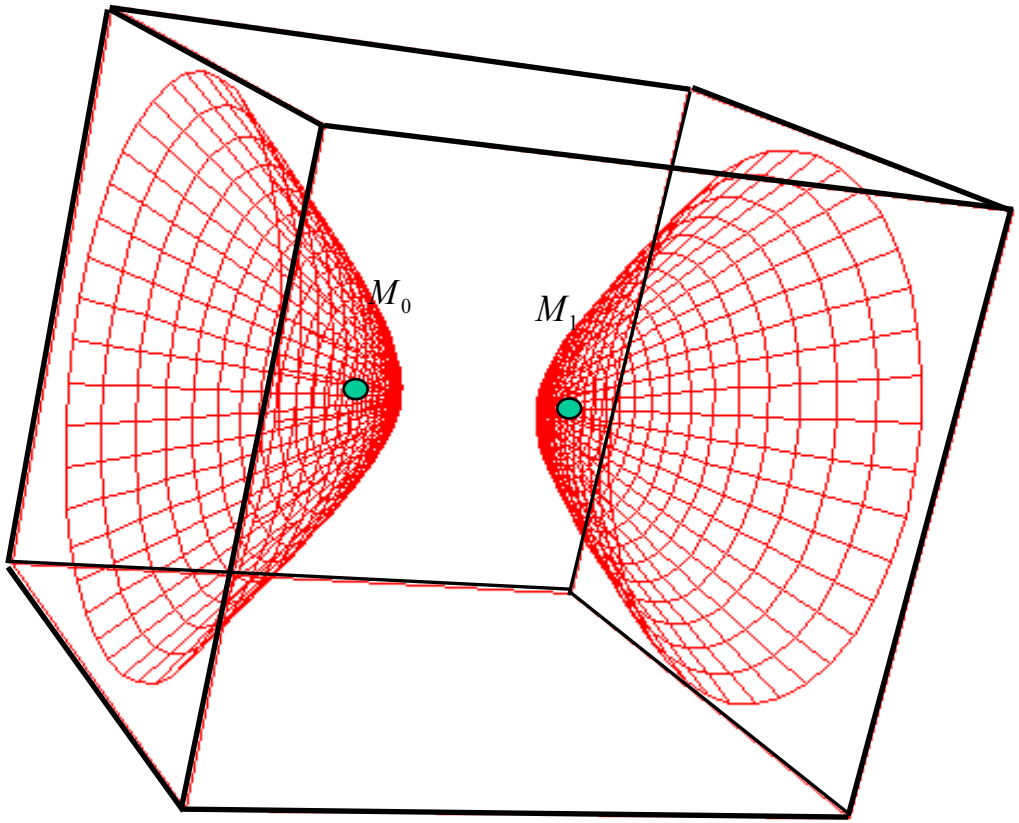
Microphone pair: TDOA-based location in 3D

Geometry extension to the 3D space leads to:

$$\tau_{01} = \frac{|\mathbf{s}-\mathbf{m}_1| - |\mathbf{s}-\mathbf{m}_0|}{c} \qquad \mathbf{s} = [s_x, s_y, s_z]$$

The locus is a hyperboloid of two sheets (left sheet for $\tau_{01}>0$, right sheet for $\tau_{01}<0$)

In a far-field situation. the sheets of the hyperboloid can by approximated by their asymptotic cones.



Use of three or more microphone pairs

Given a set of M TDOA estimates (corresponding to M microphone pairs), the best estimation of the source location **s** would be obtained ideally as the intersection of all the potential source loci.

In practice no common intersection is obtained, due to:

- errors in the TDOA estimates
- inaccuracies in the microphone coordinates
- non-ideal radiation, propagation, sensor characteristics

The solution can be found as the **best fit** with observed TDOA by minimization of an error, e.g. the least square error:

$$E(\mathbf{s}) = \sum_{i=0}^{M-1} (\hat{\tau}_i - T_i(\mathbf{s}))^2$$

$\hat{\tau}_i$ estimated delays

$T_i(\mathbf{s})$ theoretical delays if source is in $\mathbf{s}$

Or to account for the reliability of the estimates:

$$E(\mathbf{s}) = \sum_{i=0}^{M-1} \frac{(\hat{\tau}_i - T_i)^2}{\mathrm{var}(\hat{\tau}_i)}$$

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} E(\mathbf{s})$$

This optimization requires the minimization of a non-linear function. Iterative procedures or closed-form methods can be applied to efficiently approximate the exact solution.

Time Delay Estimation: Cross-Correlation

Estimation of mutual time delay between two signals $y_0(t)$ and $y_1(t)$:

Cross-correlation:

$$cc_{01}(\tau) = \int_{-\infty}^{+\infty} y_0(t) y_1(t+\tau) dt$$

in frequency domain:

$$cc_{01}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} Y_0(\omega) Y_1^*(\omega) e^{j\omega\tau} d\omega$$

It is estimated for a temporal window centered in t and length T_w

$$cc_{01}(\tau) = \int_{t-T_w/2}^{t+T_w/2} y_0(t) y_1(t+\tau) dt$$

delay estimate as peak of cross-correlation:

$$\hat{\tau}_{01} = \underset{|\tau|<d/c}{arg\,max} \left[cc_{01}(\tau) \right]$$

The peak of cross-correlation is influenced by the signals' autocorrelation.
It may be quite broad and sensitive to noise and reverberation.

A sharper peak can be obtained by prefiltering of the signals.

Generalized Cross-Correlation (Knapp-Carter'76):

$$gcc_{01}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}[G_0(\omega)Y_0(\omega)][G_1(\omega)Y_1(\omega)]^* e^{j\omega\tau}d\omega$$

$$gcc_{01}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\Psi_{01}(\omega)\cdot[Y_0(\omega)Y_1^*(\omega)]\, e^{j\omega\tau}d\omega$$

The **GCC- PHAT** (Phase Transform), corresponding to the **CSP** (Cross-power Spectrum Phase) analysis, is obtained with:

$$\Psi_{01}(\omega) = \frac{1}{|Y_0(\omega)Y_1^*(\omega)|} \Rightarrow$$

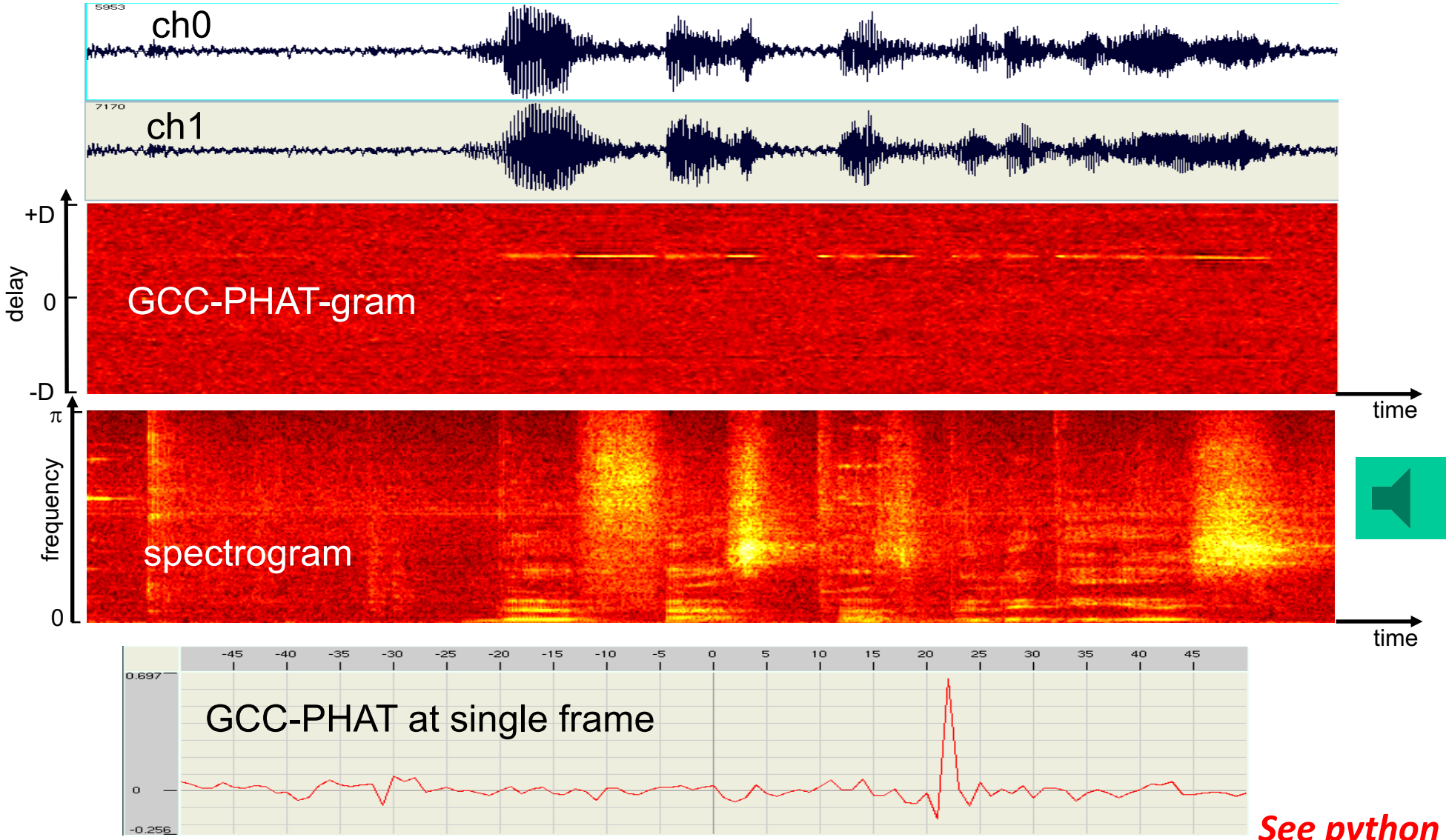
$$\Rightarrow gcc_{01}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{Y_0(\omega)Y_1^*(\omega)}{|Y_0(\omega)Y_1^*(\omega)|}e^{j\omega\tau}d\omega \Rightarrow$$

Amplitude is normalized to 1.
Only phase information is preserved!

$$\Rightarrow \hat{\tau}_{01} = \arg\max_{|\tau|<d/c}[gcc_{01}(\tau)] \Rightarrow$$

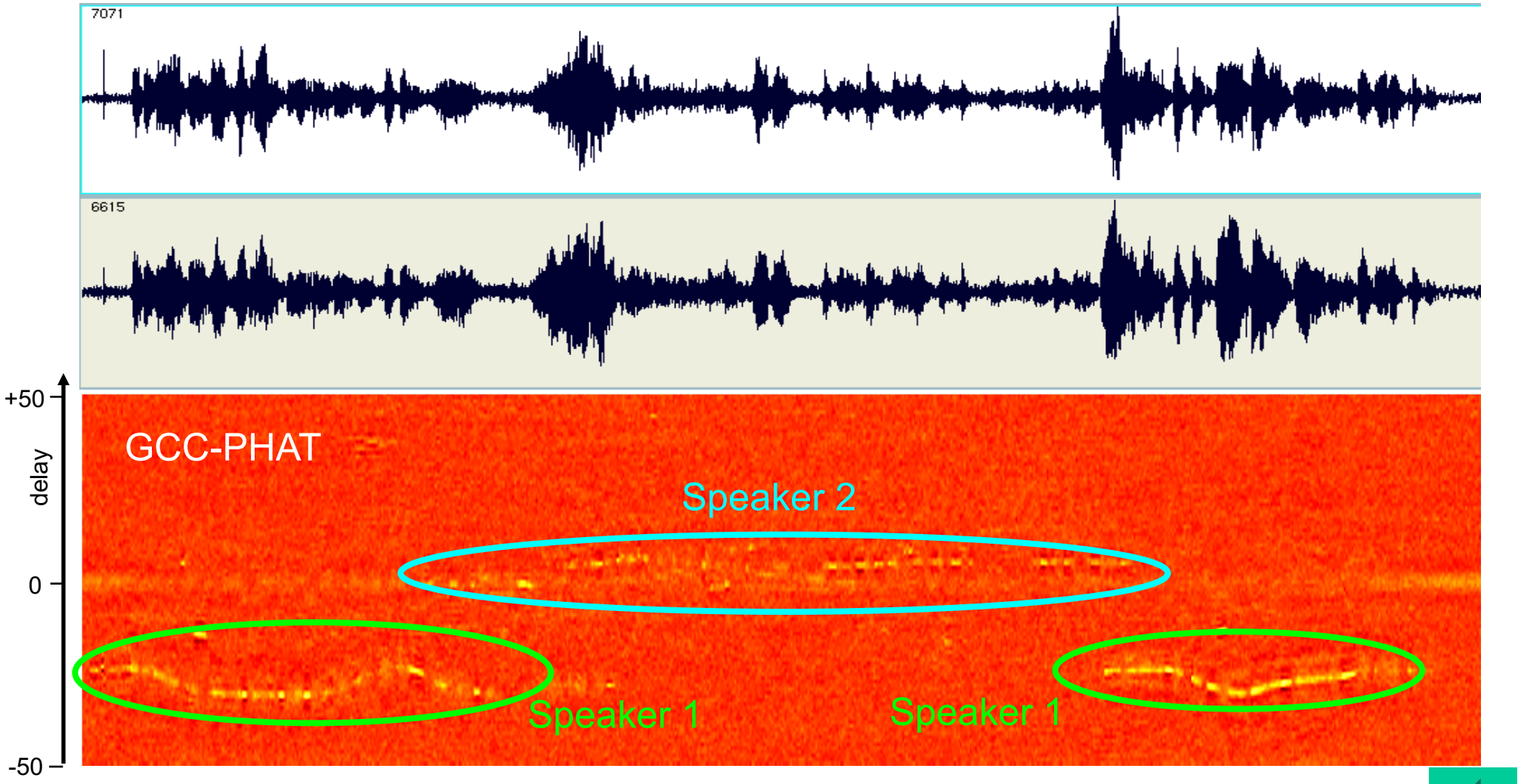
An interpolation refinement is needed to get accurate fractional delay estimates.

Application of GCC-PHAT analysis: single speaker



See python examples

Application of GCC-PHAT analysis: two speakers



Acoustic source location: common approaches and techniques

1. Indirect Methods ⇨ dual-step procedures
2. Direct Methods ⇨ single-step procedures

a. TDOA estimation
b. Apply geometry

1. Memoryless solutions ⇨ frame-by-frame independent estimation of the source position
2. Memory-based solutions ⇨ regularize a sequence of positions on a temporal interval longer than one frame

Kalman, Particle filtering etc

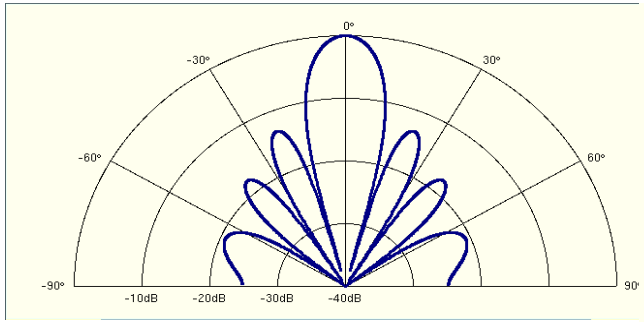
Maximization in space of a 3-D (or 2-D) Power (or Coherence) Field function

Power Field

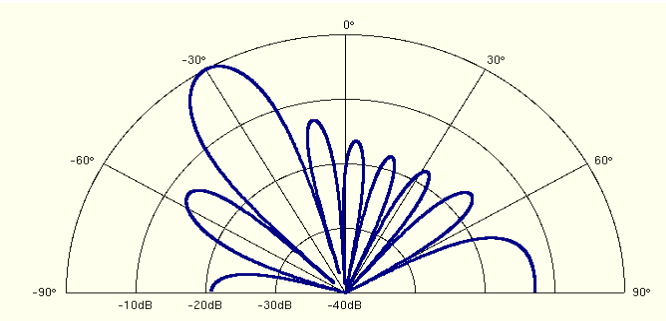
Beamforming is the modification of directivity obtainable by processing the signals of an array of sensors. Simplest approach:

Delay and sum $z(t,\mathbf{s}) = \sum_{n=0}^{M-1} y_n\left(t + T_n(\mathbf{s})\right)$ T_n=steering delays

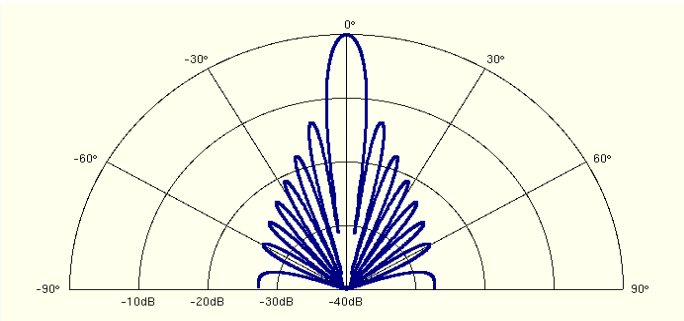
Examples with a linear microphone array:



Array steered at 0° (1kHz)



Array steered at -30° (1kHz)



Array steered at 0° (2kHz)

Beamforming can be used to "scan" the space and look for a maximum of received power. This is the Power Field (PF) approach [Alvarado 1990], also more recently called as SRP (Steered Response Power).

Drawbacks:

- computationally expensive
- no strong global peak
- highly dependent on the spectral content of the signals

Global Coherence Field

A hybrid approach conjugates the advantages of TDOA method and Power Field.

Given a set M_P of microphone pairs the Global Coherence Field* (GCF) [Omologo-Svaizer 1993, 1997] is computed at time instant t as:

$$GCF(t,\boldsymbol{s}) = \frac{1}{M_P} \sum_{(i,k)\in\{M_P\}} gcc_{ik}\left(t, \delta_{ik}(\boldsymbol{s})\right)$$

where $\delta_{ik}(\boldsymbol{s})$ denotes the theoretical delay for the (i,k) microphone pair having assumed that the source is in position $\boldsymbol{s}$

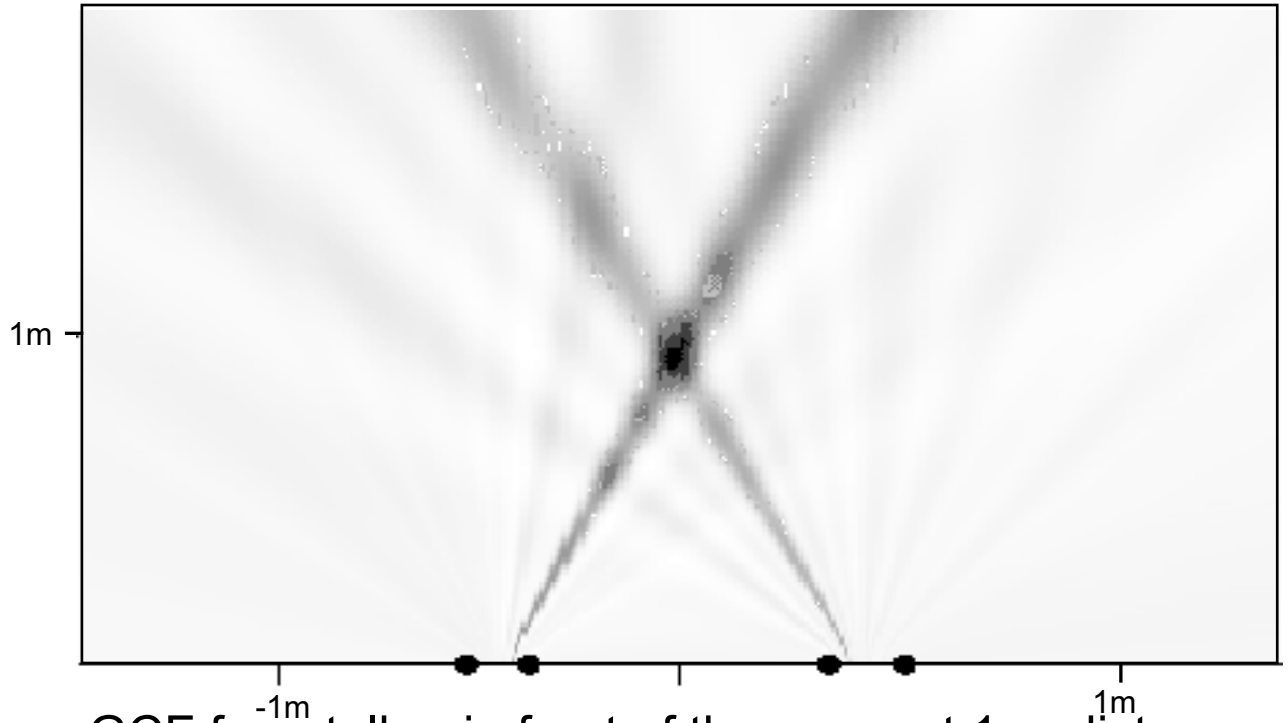
$$\Rightarrow \hat{\boldsymbol{s}}(t) = \arg\max_{\boldsymbol{s}} GCF(t,\boldsymbol{s})$$

Advantages: The GCF provides a sharper peak than Power Field, with a consequent decreased sensitivity to noise and reverberation.
Moreover, it is a direct single-step method.

*** [Brandstein-Ward 2001] uses the term SRP-PHAT to indicate the above described technique for GCF computation. In practice, it is the same method.**

GCF-based acoustic maps

Example of Global Coherence Field accounting for the contributes of the CSP functions related to two microphone pairs:

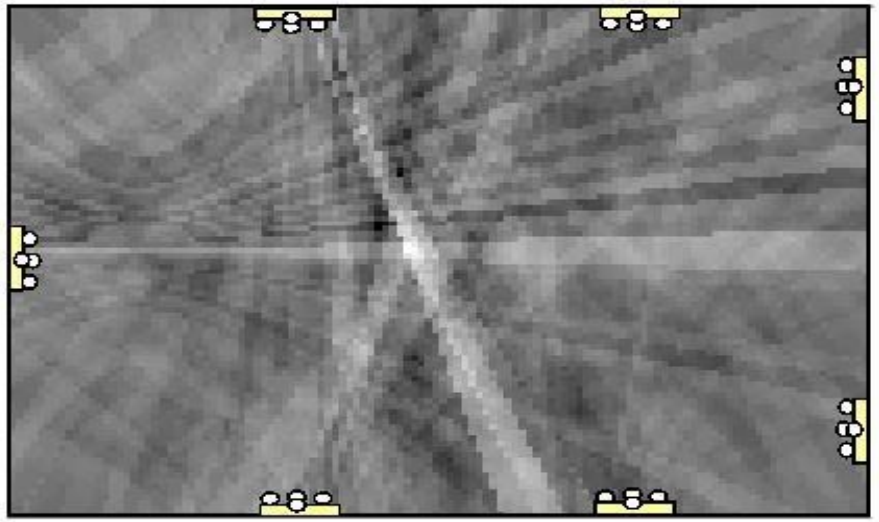


GCF for a talker in front of the array, at 1 m distance

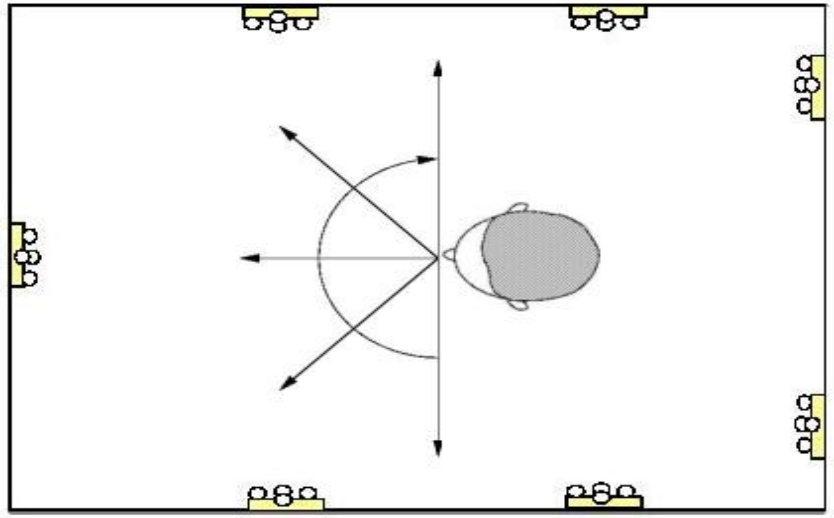
Note that GCF preserves all the information expressed by the CSP functions, not only the bearing directions associated to main peaks.

See python examples

Use of GCF to estimate Head Orientation



Example of 2D GCF in a real room



According to head orientation the contribution of the various micro-phone pairs have different strength

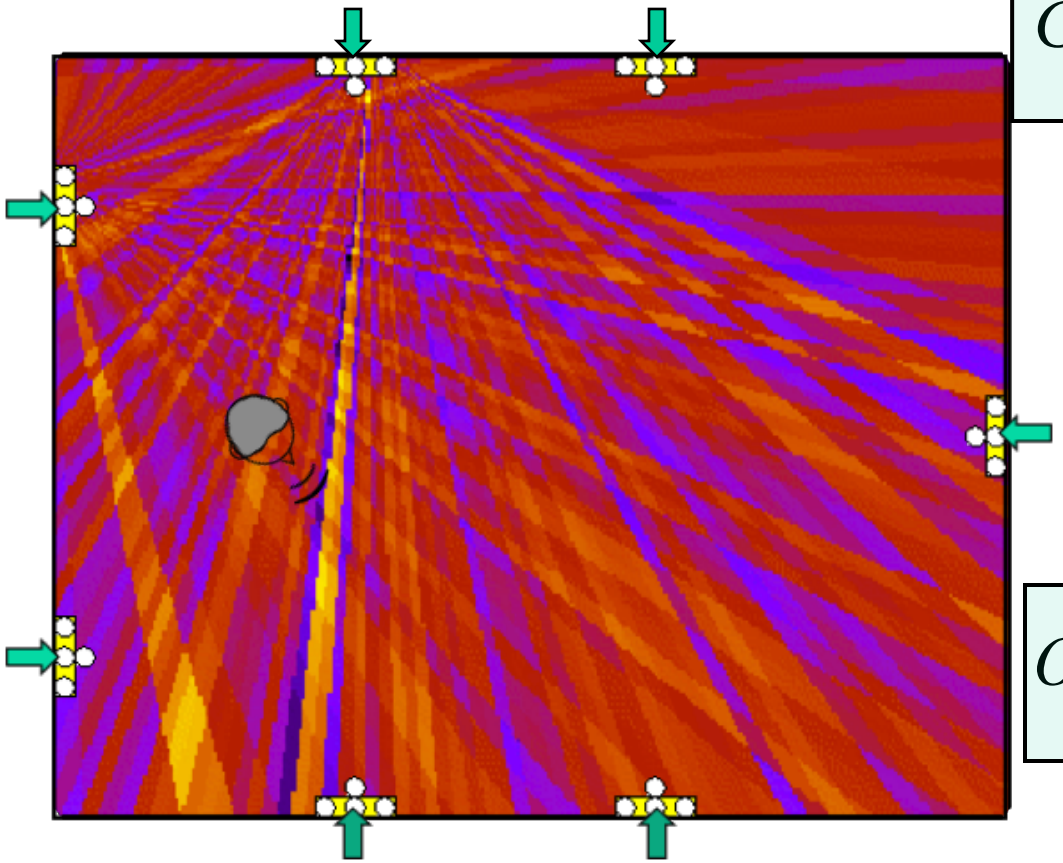
➡ The relative variations of GCF around the source position are clues to deduce source orientation

➡ The audio map of GCF can be exploited to derive information about talker orientation

➡ ***Oriented Global Coherence Field*** (one GCF for each direction)

From GCC-PHAT to Global Coherence Field (GCF) and Oriented GCF

$$gcc_{01}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{Y_0(\omega) Y_1^*(\omega)}{\left| Y_0(\omega) Y_1^*(\omega) \right|} e^{j\omega\tau} d\omega$$



Given M_P microphone pairs:

$$GCF(t, \mathbf{s}) = \frac{1}{M_P} \sum_{(i,k) \in \{M_P\}} gcc_{ik}\left(t, \delta_{ik}(\mathbf{s})\right)$$

$\delta_{ik}(\mathbf{s})$ = theoretical delay

$$\hat{\mathbf{s}}(t) = \arg\max_{\mathbf{s}} GCF(t, \mathbf{s})$$

Location

for every direction *j*:

$$OGCF_j(t, \mathbf{s}) = \sum_{l=0}^{L-1} GCF_{\Omega_l}(t, Q_l) w(\theta_{lj})$$

$$\hat{\mathbf{s}}(t) = \arg\max_{\mathbf{s}, j} OGCF_j(t, \mathbf{s})$$

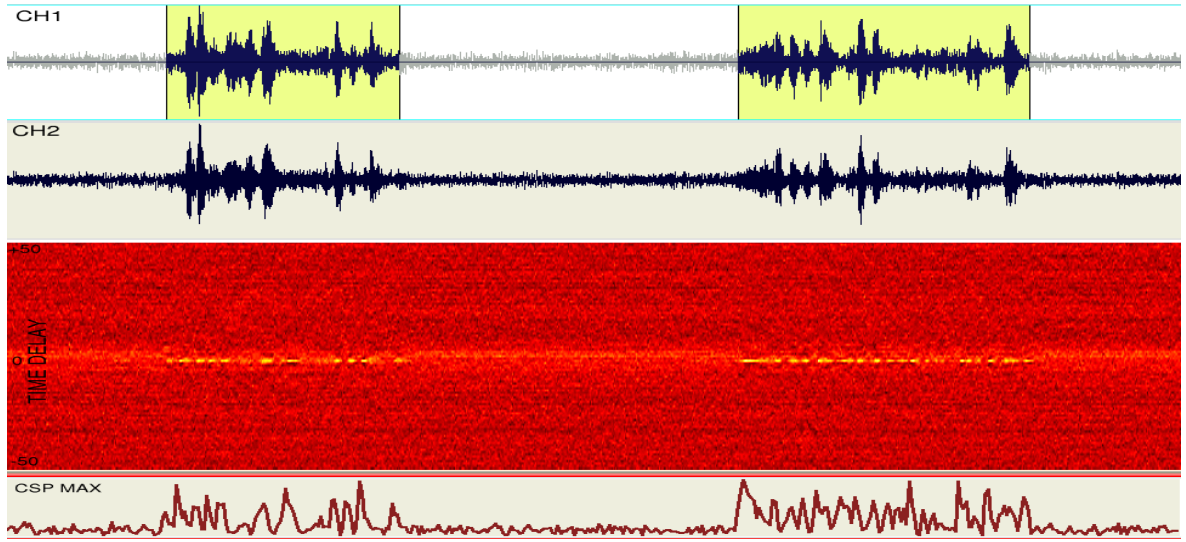
Location+Orientation

More details on GCF in [De Mori 1998].
As for OGCF, see [Brutti et al. 2005].
Extended to multiple speaker location, see [Brutti et al. 2010].

Speech Activity Detection (SAD) for Speaker Location and Tracking

- In a real noisy and reverberant environment, SAD is a very challenging task!
- Many techniques proposed in the literature, but still a very open research issue for DSR
- In a real application, a speaker location and tracking system is also characterized by its capabilities to produce in real-time position estimates only when a speaker is active, i.e, reducing false alarms and deletions.
- The peaks of the CSP or of GCF and OGCF functions are suitable features for speech activity detection algorithms [Armani et al. 2003, Brutti et al. 2005].

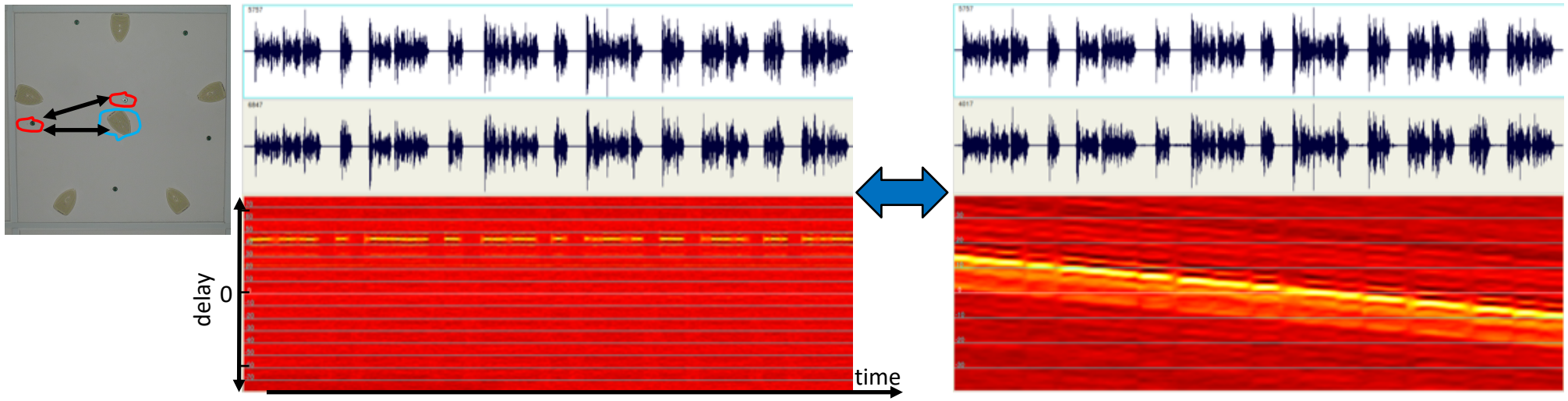
In the following example, the speaker was at 3 m distance from the microphones:



Ad-hoc microphone arrays: clock drift

- Lack of information on device position and clock synchronization
  - Passive microphone position self-calibration [Plinge 2016]
  - Blind estimation of sampling frequency mismatch [Miyabe 2015]
  - Very challenging topics under real-world conditions, with WASNs, reverberation, multiple near-field directive sources, non-omni microphones

Example of DIRHA_English: GCC-PHAT-grams with synch. vs asynch. microphones

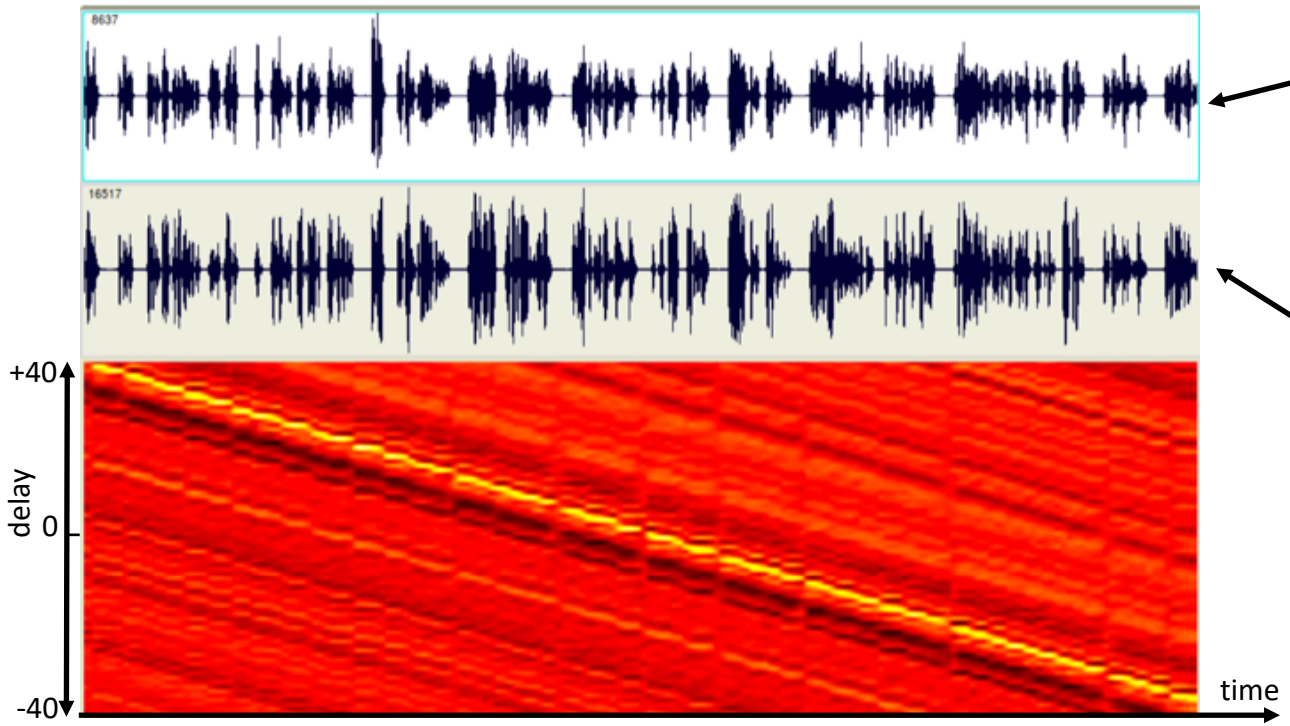


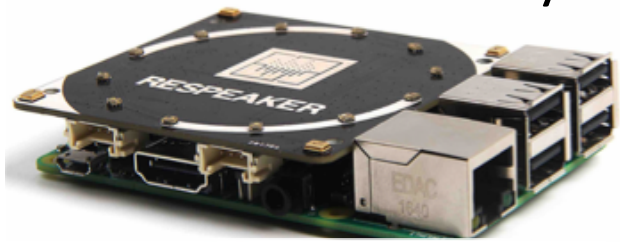
Possible target: Cooperative neural framework solving the problem in a blind way

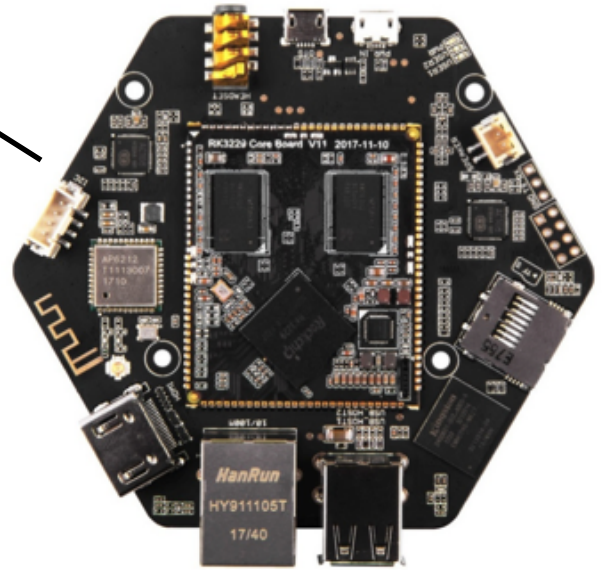
A. Plinge et al., "*Acoustic Microphone Geometry Calibration*", in IEEE Signal Processing Magazine, July 2016.
S. Miyabe et al., "*Blind compensation of interchannel sampling frequency mismatch for ad-hoc microphone array based on maximum likelihood estimation*", Signal Processing 107 (2015), pp. 185-196.

CAV3D2 corpus based on ad-hoc microphone arrays

- Conversations in a room with about 0.7 s reverberation time
- Use of three arrays
- Annotation under way for speaker localization purposes
- Not clear if by the end of June we will have the annotation necessary for JSALT

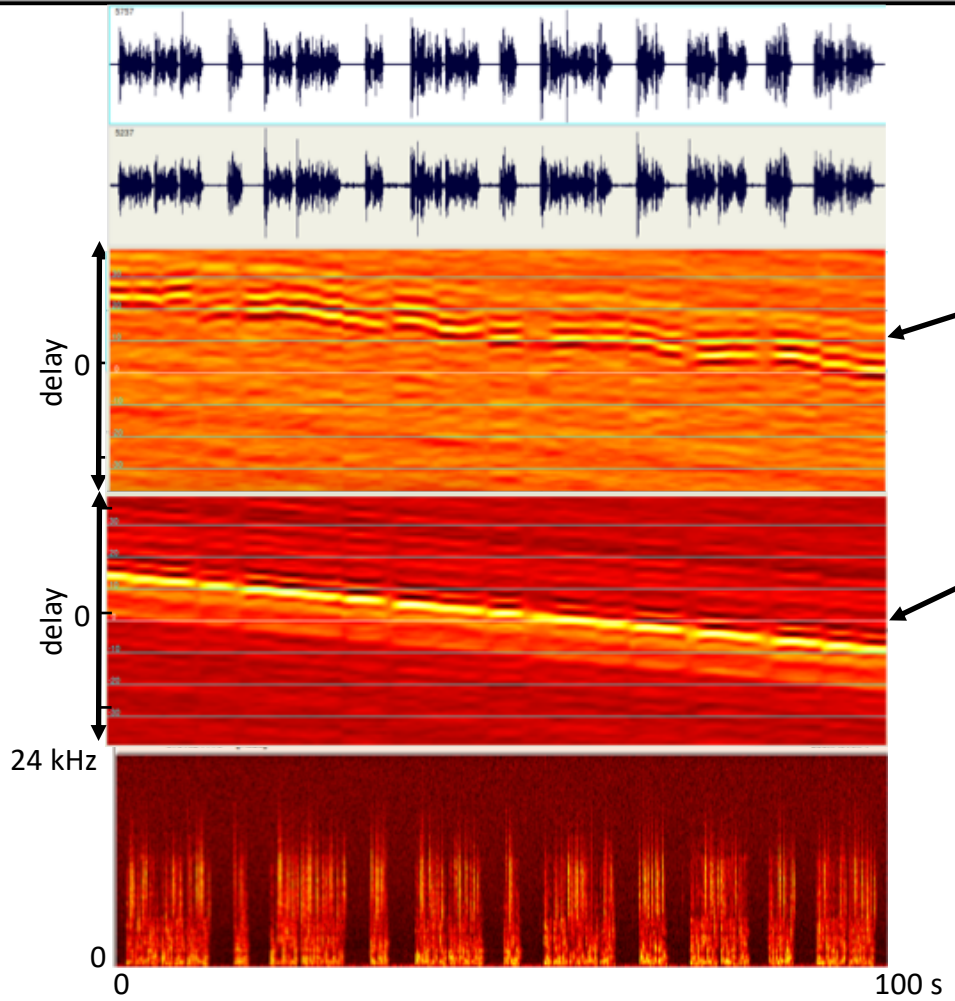


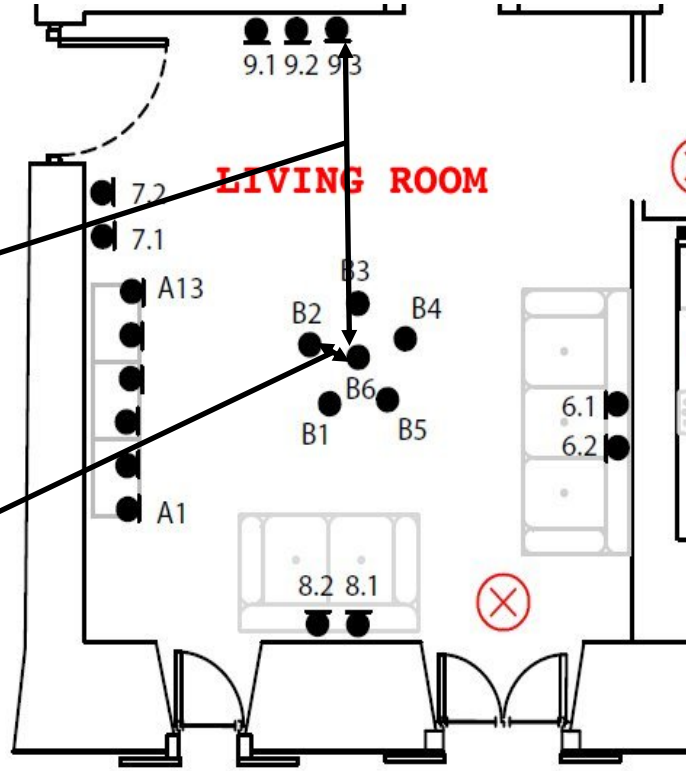




Drift of 74 samples in 63s, at 16kHz (i.e. about 73.5 ppm)

Clock mismatch in DIRHA-English recordings





Drift of about 4.8 ppm

References

For an overview of the topic, and for most recent references, see:

- P. Pertilä, A. Brutti, P. Svaizer, and M. Omologo, *Multichannel source activity detection, localization, and tracking,* Chapter 4 in: E. Vincent; T. Virtanen; S. Gannot (Eds), *Audio source separation and speech enhancement*, Wiley, 2018.

Some additional older references in the field:

• H. Kuttruff, *"Room Acoustics",* Elsevier Applied Science, (3rd edition) 1991.

• J. Blauert, *"Spatial Hearing",* MIT Press, (Revised Edition) 1997.

• D. Johnson, D. Dudgeon, *"Array Signal Processing – Concepts and Techniques",* Prentice Hall, 1993.

• M. Omologo, P. Svaizer, R. De Mori, *"Acoustic transduction",* ch. 2 of *"Spoken dialogues with computers",* R. De Mori ed., Academic Press, 1998.

• M. Brandstein and D. Ward eds, *''Microphone Arrays''*, Springer Verlag, 2001.

• Y. Huang, J. Benesty, G.W. Elko, *"Microphone arrays for video camera steering",* ch. 11 of *"Acoustic signal processing for telecommunication",* S.L. Gay and J. Benesty eds., Kluwer, 2000.

• Y. Huang, J. Benesty, "*Audio Signal Processing for Next-Generation Multimedia Communication Systems*", Chapters 8 and 9, Kluwer 2004

References

• C.H. Knapp, G.C. Carter, *"The generalized correlation method for estimation of time delay",* IEEE Trans. on ASSP, vol. 24, pp. 320-327, 1976.

• J.B. Allen, D.A. Berkley, "*Image method for efficiently simulating small-room acoustics*", JASA, vol. 65, pp. 943-950, 1979.

• V. M. Alvarado, "*Talker localization and optimal placement of microphones with a linear microphone array using stochastic region contraction"*, PhD Thesis, Brown University, 1990.

• M. Omologo and P. Svaizer, "*Use of the Cross-power Spectrum Phase in Acoustic Event Localization*", ITC-irst Tecnical Report #9303-13, March 1993.

• J. L. Flanagan, A. Surendran, E. Jan, "*Spatially selective sound capture for speech and audio processing*", Speech Communication, vol. 13, pp. 207-222, 1993.

• M. Omologo, P. Svaizer, *"Acoustic event location using a Crosspower-Spectrum Phase based technique",* Proc. Of IEEE ICASSP 1994, pp.273-276.

• B. Champagne, S. Bédard, A. Stéphenne, *"Performance of time delay estimation in the presence of room reverberation",* IEEE Trans. on SAP, vol. 4, pp. 148-152, 1996.

• M. Omologo, P. Svaizer, "*Use of the crosspower-spectrum phase in acoustic event location*", IEEE Trans. on SAP, vol. 5, pp. 288-292, 1997.

• H. Wang, P. Chu, "*Voice source location for automatic camera pointing system in videoconferencing*", in Proc. of IEEE ICASSP 1997, pp. 187-190.

References

•P. Svaizer, M. Matassoni, M. Omologo, "*Acoustic source location in a three-dimensional space using crosspower spectrum phase*", in Proc. of IEEE ICASSP 1997, pp. 231-234.

• H. Buchner et al.,"*Simultaneous localization of multiple sound sources using blind adaptive MIMO filtering"*, Proc. of ICASSP, vol. III, pp. 97-100, 2005

• H. Teutsch, W. Kellermann, " *EB-ESPRIT: 2D localization of multiple wideband acoustic sources using eigen-beams*" ,Proc. of ICASSP 2005

•M. Omologo et al., "*Speaker Localization in CHIL lectures: Evaluation Criteria and Results*", MLMI'05, Springer Lecture Notes in Computer Science vol. 3869, 2006.

•D. Zotkin et al., "*Multimodal 3-D tracking and event detection via the particle filter*", IEEE Workshop on Detection and Recognition of Event in Video, 2001

• N. Strobel, S. Spors, and R. Rabenstein, "*Joint audio-video object localization and tracking*", IEEE Signal Processing Magazine, vol. 18,Jan. 2001.

• D.B. Ward and R.C. Williamson, "*Particle filter beamforming for acoustic source localization in a reverberant environment*", Proc. of ICASSP 2002.

• J. Chen, J. Benesty, Y. Huang, *"Robust time delay estimation exploiting redundancy among multiple microphones",* IEEE Trans. on SAP, vol. 11, pp. 549-557, 2003.

• T.G. Dvorkind and S. Gannot, "*Speaker Localization exploiting spatial-temporal information*", IEEE Workshop on Ac. Echo and Noise control, September 2003.

References

• L. Armani, M. Matassoni, M. Omologo, P. Svaizer, "*Use of a CSP-based voice detector for distant-talking ASR*", Proc. of EUROSPEECH, Geneva, Switzerland, September 2003.

• E. Lehmann et al., "*Experimental comparison of particle filtering algorithms for acoustic source localization in reverberant room*", Proc. of ICASSP 2003.

• D. B. Ward et al., "*Particle filtering algorithms for tracking an acoustic source in a reverberant environment*", IEEE Trans. on SAP, vol. 11, n.6, pp. 826-836, 2003.

• S. Doclo and M. Moonen, "*Robust Adaptive Time Delay Estimation for Speaker Localization in Noisy and reverberant Acoustic Environments"*, EURASIP Journal on Applied Signal Processing, vol. 11, pp. 1110-1124, 2003.

•J. Benesty et al., "*Time-delay estimation via Linear Interpolation and cross-correlation*", IEEE Trans. on Speech and Audio Processing, vol. 12, n. 5, 2004.

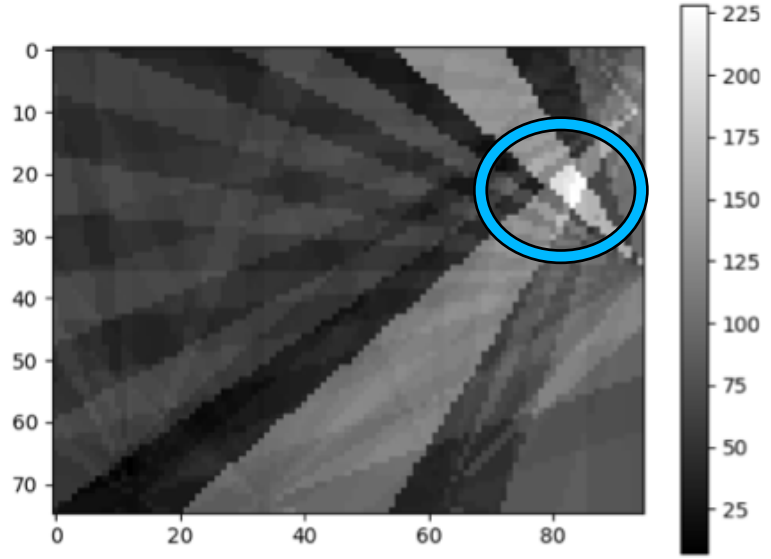
• A. Brutti, M. Omologo, P. Svaizer, "*Oriented global coherence field for the estimation of the head orientation in smart rooms equipped with distributed microphone arrays*", Proc. of Interspeech 2005.

A. Brutti, M. Omologo, P. Svaizer, "*Multiple Source Localization Based on Acoustic Map De-Emphasis* ", EURASIP Journal on Audio, Speech, and Music Processing, Vol. 2010 Issue 1, December 2010

Examples of GCF from segments of the CHiME5 like artificial corpus

```
    <Phones>  AH1 DH ER0 B ER1 G AH0 N D IY0 Z AA1 R </Phones>
  </Activity>
  <Activity>
    <ActivityIndex> 23 </ActivityIndex>
    <Sp_Index> 0 </Sp_Index>
    <SourceIndex> 23 </SourceIndex>
    <StartTimeSample> 1226229 </StartTimeSample>
    <EndTimeSample> 1302549 </EndTimeSample>
    <AmpFactor> 0.93 </AmpFactor>
    <Type>Speech</Type>
    <RootDir>Librispeech/dev-clean</RootDir>
    <SpeakerDir> 6313 </SpeakerDir>
    <SubDir> 76958 </SubDir>
    <UtteranceIndex> 4 </UtteranceIndex>
    <UtteranceSegm> 0 </UtteranceSegm>
    <StartSegmSample> 7040 </StartSegmSample>
    <EndSegmSample> 83360 </EndSegmSample>
    <Words>  the pony did most of it admitted the lad i just gave
 his head and that's all there was to it </Words>
    <Phones>  DH AH0 P OW1 N IY2 D IH1 D M OW1 S T AH0 V IH0 T AH0
M IH1 T AH0 D DH AH0 L AE1 D sp AY1 JH IH0 S T G EY1 V IH0 M HH IH1
HH EH1 D AH0 N D DH AE1 T S AO1 L DH EH1 R W AH1 Z T UW1 IH0 T </Ph
s>
```

```
      <X> 2345 </X>
      <Y> 8428 </Y>
      <Z> 1500 </Z>
      <Or_Az> 302 </Or_Az>
      <Or_El> 89 </Or_El>
    </SourceLabXYZ>
    <SourceLabXYZ>
      <Index> 23 </Index>
      <Sp_Index> 0 </Sp_Index>
      <X> 2382 </X>
      <Y> 8295 </Y>
      <Z> 1500 </Z>
      <Or_Az> 50 </Or_Az>
      <Or_El> 63 </Or_El>
    </SourceLabXYZ>
    <SourceLabXYZ>
      <Index> 24 </Index>
      <Sp_Index> 1 </Sp_Index>
      <X> 2025 </X>
      <Y> 8565 </Y>
      <Z> 1500 </Z>
      <Or_Az> 205 </Or_Az>
      <Or_El> 97 </Or_El>
```




```
    </Activity>
    <Activity>
      <ActivityIndex> 14 </ActivityIndex>
      <Sp_Index> 1 </Sp_Index>
      <SourceIndex> 14 </SourceIndex>
      <StartTimeSample> 731415 </StartTimeSample>
      <EndTimeSample> 817015 </EndTimeSample>
      <AmpFactor> 0.73 </AmpFactor>
      <Type>Speech</Type>
      <RootDir>Librispeech/dev-clean</RootDir>
      <SpeakerDir> 2803 </SpeakerDir>
      <SubDir> 154320 </SubDir>
      <UtteranceIndex> 1 </UtteranceIndex>
      <UtteranceSegm> 1 </UtteranceSegm>
      <StartSegmSample> 107040 </StartSegmSample>
      <EndSegmSample> 192640 </EndSegmSample>
      <Words>  but it grieved him that his companions should have to
 suffer so much discomfort </Words>
      <Phones>  B AH1 T IH0 T G R IY1 V D HH IH1 M sp DH AH0 T HH IH
0 Z K AH0 M P AE1 N Y AH0 N Z SH UH1 D HH AE1 V T UW1 S AH1 F ER0 S OW
1 M AH1 CH D IH0 S K AH1 M F ER0 T </Phones>
    </Activity>
    <Activity>
```

```
      <X> 5813 </X>
      <Y> 4504 </Y>
      <Z> 1500 </Z>
      <Or_Az> 17 </Or_Az>
      <Or_El> 83 </Or_El>
    </SourceLabXYZ>
    <SourceLabXYZ>
      <Index> 14 </Index>
      <Sp_Index> 1 </Sp_Index>
      <X> 5611 </X>
      <Y> 4629 </Y>
      <Z> 1500 </Z>
      <Or_Az> 313 </Or_Az>
      <Or_El> 108 </Or_El>
    </SourceLabXYZ>
    <SourceLabXYZ>
      <Index> 15 </Index>
      <Sp_Index> 2 </Sp_Index>
      <X> 5425 </X>
      <Y> 4627 </Y>
      <Z> 1500 </Z>
      <Or_Az> 288 </Or_Az>
      <Or_El> 98 </Or_El>
```