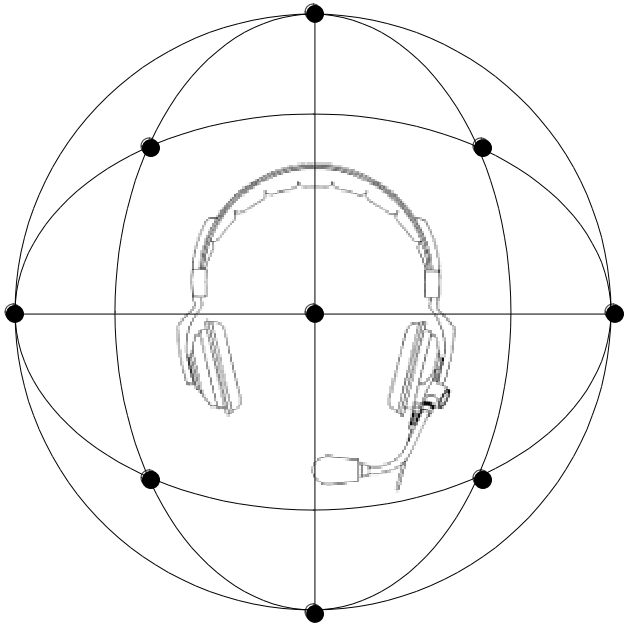
Multimodal Interfaces

lecture 02: Auditory User Interfaces



Univ.Prof. Martin Kaltenbrunner

Interface Culture Lab, Kunstuniversität Linz

++ introduction: why sound?

+ neglected role of sound in HCI --> like silent movies
+ auditory perception is not generally less important,
  it just has a different function than visual perception

+ visual concepts are powerful, but in many cases inadequate
  (permanent attention, limited space, …)

+ highly immersive augmented reality:
  synthetic sound really exists - compared to visual projections

+ technologically advanced synthesis techniques
  it’s cheap and easy

++ definition

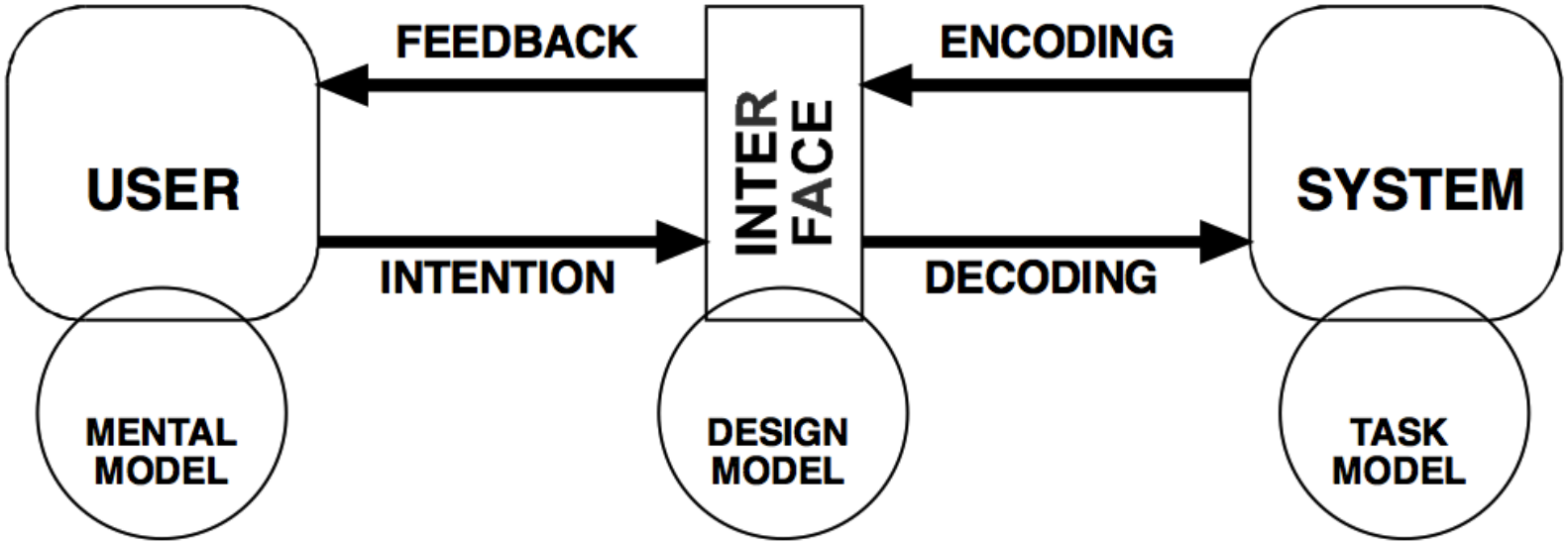
+ AUI // Auditory User Interface

"the use of non-speech sound to convey information"

--> acoustically rendered interface

- natural (recorded) & synthesized sound
- speech synthesis can be added
  --> but AUIs are more than speech systems only !!!
- GUIs can be enhanced with AUI techniques -> mixed mode
- does not define user control, could be keyboard, voice, tracking

++ general HCI model



++ questions and remarks to think about

- how many different dimensions and variables are available and usable for display in a sonic universe?

- what are the major differences from the way we visually & acoustically perceive our surroundings?

- hearing is a passive process, we can easily listen to the radio while being occupied with other tasks. vision requires active attention!

- we only can see those objects in our visual field, while we can hear everything which surrounds us, we even can hear things we can’t see!

- auditory perception is linear in time: sound objects exist in a certain moment of time, but can be perceived over space.

- visual perception is linear in space: visual objects exist in a certain location in space, but can be perceived over time.

++ AUI applications

+ mobile devices

- phones, PDAs, wearables, lifestyle devices
  - insufficiently small screens & tiny keyboards
  - increasing processing power & features
- remote access/control

+ embedded devices

- consumer home electronics
- industrial applications, monitoring

+ desktop systems

- GUI extension – acoustic feedback
- hands free operation (control, dictation)

+ accessibility

- interface sonification for visually disabled users

+ military

- auditory display for pilots

++ accessibility

+ screen readers

auralize/speak GUI (graphical) or CLI (command line)

using software/hardware speech synthesisers.

speech-enabled web-browsers

+ emacspeak

complete auditory desktop by T.V. Raman (a blind scientist)

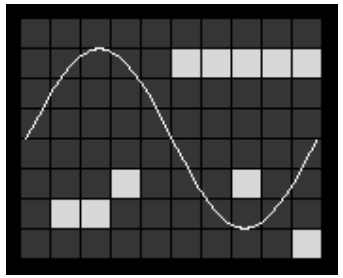
voice & audio feedback for GNU/emacs

keyboard shortcuts

+ experimental systems

video/image sonification

orientation (bat/sonar) systems



++ areas

+ voice systems

speech/voice driven human-computer interaction

menus – lists – forms – dialogs

+ auditory feedback

enhancing interaction with acoustic feedback

multi-modal interfaces

+ data sonification, interactive sonification

mapping data to acoustic parameters

auditory analogy to visualisation

+ acoustic monitoring

no active visual attention necessary

perception of changes in audio streams

events call attention: no sound, additional sound, different sound

example: peep network auralizer

++ more on sonification

especially suitable for time-varying data

comparing simultaneous data streams (left/right ear)

identification of patterns in auralised data streams (timbre)

dimensions of sound:

physical: frequency, amplitude, phase, spectral parameters

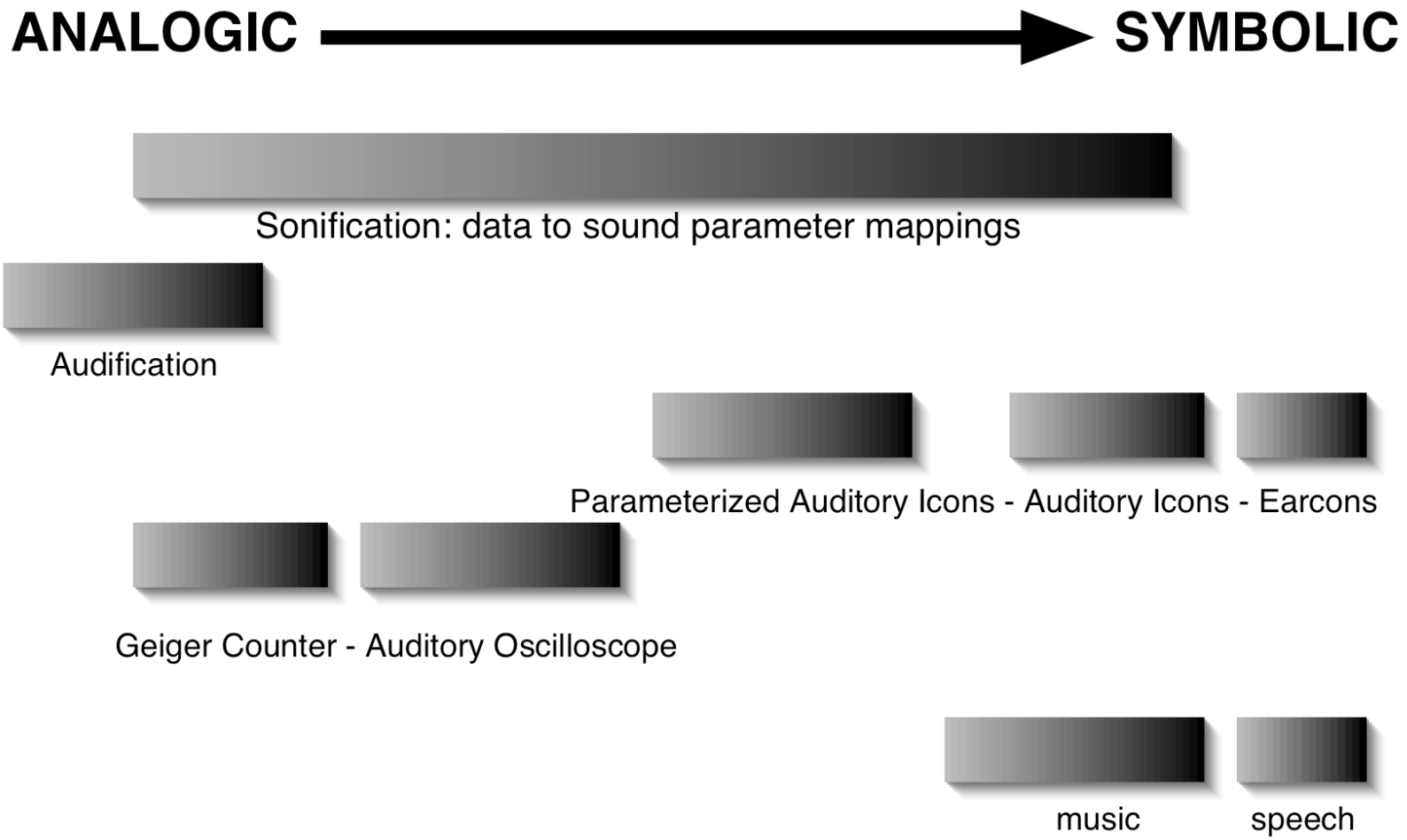
perceptual: pitch, loudness, timbre, rhythm

mapping: applying data features to sound features

sonification examples: EEG data 

audification examples: earthquake data 

++ auditory information representation



after G. Kramer: An introduction to Auditory Display

++ elements

+ auditory icons

- short sound samples (natural or synthesised), carrying symbolic meaning about the associated object, task or event
- parameterized auditory icons are synthesized symbolic sounds allowing full control over their appearance by modification of the synthesis parameters
- common for enhancing GUI environments (sonic finder)

+ earcons

- short harmonic sequences with a dedicated characteristics, such as a common chord
- allows the navigation within complex hierarchies, such as the browsing through a tree structure (a mobile phone menu for example)

example: menu tree navigation (Brewster)

++ multiple audio streams

+ cocktail party effect

for the simultaneous display of applications/sources

cocktail party effect allows source segregation

additional techniques to support source segregation

- sound effects

support the distinction of various sources, spaces, categories

apply quality (urgency etc.)

- spatial audio

placement of simultaneous streams/events in auditory space

headphone (HRTF) or speaker (panning) rendering

++ AUI elements

+ voice

synthesis

standard parameters: gender, age, speed

additional features: emotion, expressiveness, singing

natural language generation

(recognition)

command & control -- dictation

voice (non-speech) control

speaker authentication & verification

natural language analysis

+ music

ready made recordings: similar function like in movies

algorithmic composition: automatically generates music from context

use of musical knowledge can improve aesthetical appearance

++ AUI elements

+ voice

synthesis

standard parameters: gender, age, speed

additional features: emotion, expressiveness, singing

natural language generation

(recognition)

command & control -- dictation

voice (non-speech) control

speaker authentication & verification

natural language analysis

+ music

ready made recordings: similar function like in movies

algorithmic composition: automatically generates music from context

use of musical knowledge can improve aesthetical appearance

++ sonic/auditory widgets

+ widget

'any small device whose name you have forgotten or do not know'

elementary user interface components

sonic widget: acoustically enhanced GUI widgets

auditory widget: elementary auditory user interface component

+ reusable libraries

GUI libraries (Swing, MFC, GTK) --> easier for developers

+ general hear & feel

easier to recognize / operate for the users

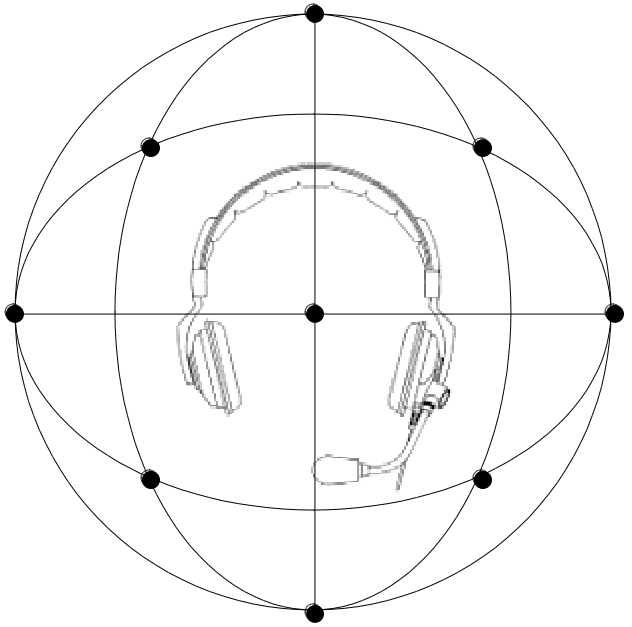
+ examples

dialogs, browser, forms, editor, progress

sample: auditory progress bar (Brewster)

Multimodal Interfaces

lecture 02b: voice technology



Univ.Prof. Martin Kaltenbrunner

Interface Culture Lab, Kunstuniversität Linz

++ speech recognition: word spotting

+ limited vocabulary

defines a set of single word commands and actions

e.g. speaking a name to a mobile for calling

+ simple speech recognition techniques

Dynamic Time Warping (DTW)

+ requires user-training

user records word-samples for recognition

+ implemented in hardware

chips for mobile phones etc.

can store a limited number of words

++ speech recognition: command & control

+ rule-grammar

the application already predefines what the user input will be

predefined actions will be triggered

user utterances out of context are not considered

+ user independent

should work with general voice models

does not require special training of a user-model

+ speech recognition methods

Hidden Markov Models

Neural Networks

++ speech recognition: continuous speech

+ text dictation

recognition of continuously spoken unknown text

free user input within interactive applications

+ user dependent

recognition rate improves with training

user reads some predefined texts to train user-model

+ dictionaries

up to 200.000 words in current products

statistical models on word probability & context

unknown words (fashion words, compounds) can not be recognized

users can analyze and add new words to dictionary

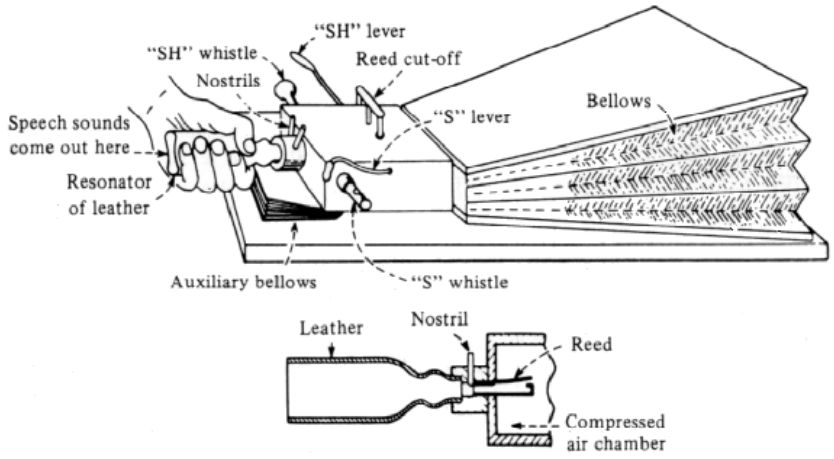
+ specialized dictionaries

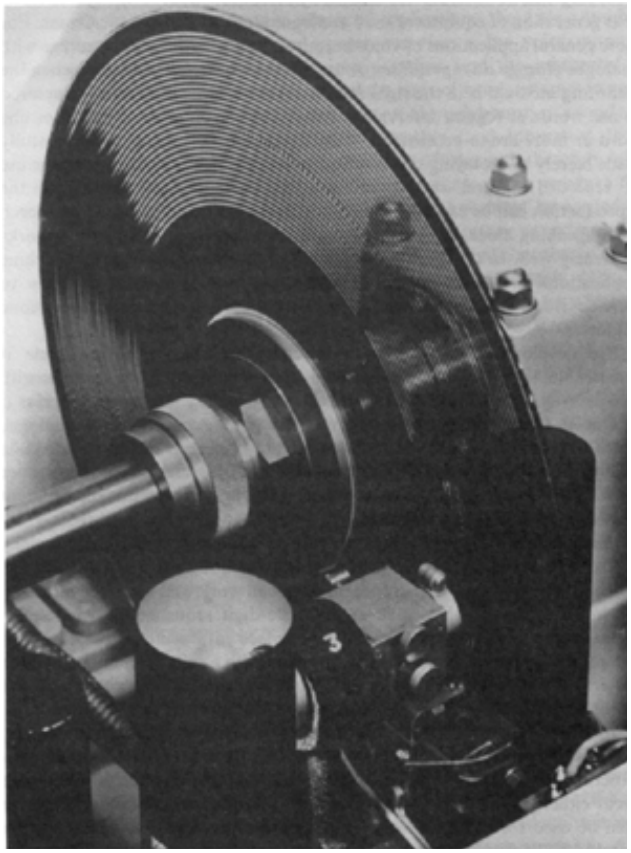
lawyers, medics etc.

professionals use special and limited vocabulary

++ speech synthesis: historical aspects

Kempelen´s speaking machine
from the 18th century





BT talking clock from 1936

++ vocoder/voder 1940

The Vocoder (Voice Operated reCorDER) developed by Homer Dudley, was a composite device consisting of an analyzer and an artificial voice. The analyzer detected energy levels of successive sound samples measured over the entire audio frequency spectrum via a series of narrow band filters.

The synthesizer reversed the process by scanning the data from the analyzer and supplying the results to a feedback network of analytical filters energized by a noise generator to produce audible sounds.

The fidelity of the machine was limited, the machine was intended as a research machine for compression schemes to transmit voice over copper phone lines.

++ speech synthesis: concatenation

+ fast & easy: recorded word samples

e.g. talking clock

pros: very good quality

cons: not very flexible, large databases

+ better approach: record and process speech

isolate phonemes from recorded speech

single phonemes insufficient, transitions do not sound natural

diphone: isolate phonemes with transitions to neighbours

44 phonemes --> 1600 diphones

pros: good quality, flexible

cons: large databases for each voice (10M)

++ speech synthesis: physical models

+ simple LPC model

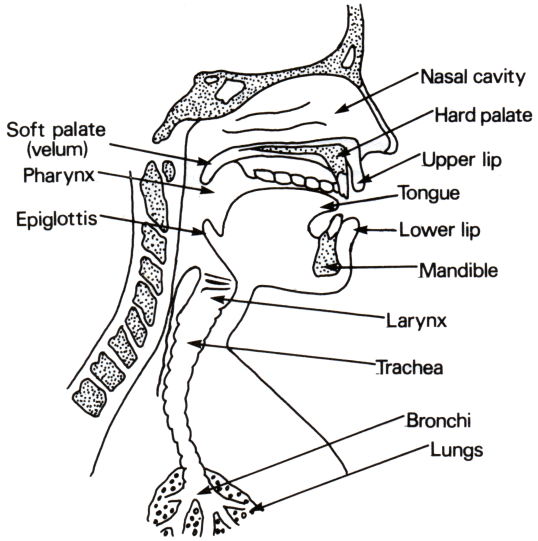
voiced/unvoiced excitation

human vocal tract filter

pros: simple, efficient

very flexible, small footprint

cons: moderate quality



+ formant synthesis

voiced excitation

modelling 3-5 voice formants

with-band pass resonators

+ other methods

- sinusoidal modelling
- mixed synthesis, combination of various methods

++ speech synthesis: higher level

+ text processing

pre-processing of abstract data necessary

numbers: 123 -> one hundred and twenty three

dates: 11.04. -> eleventh of April

+ pronunciation

phonetic transcription of plain text

rules for correct pronunciation of words

dictionaries for exceptional words

+ prosody

e.g. for end of sentences, question marks, emphasis etc.

changes of pitch contour, loudness, speed etc.

adds expressiveness: singing, shouting

++ singing voice synthesis

+ first examples

Max Mathews synthesized the first song in 1961

“Daisy Bell”, was actually cited in Kubrick’s 2001

+ utabara, online synthesizer

http://www.utabara.com/

+ flinger, free software

http://speech.bme.ogi.edu/tts/flinger/

based on festival, MIDI input

+ MTG “daisy” -> Yamaha Vocaloid

http://www.vocaloid.com/

high quality singing voice synthesizer

takes score+lyrics as input

-> manual editing of expressiveness (vibrato …)

Very popular in Japab: Hatsune Miku

++ voice control

+ voice feature extraction

simple: loudness, etc.

vocal tract: vowel recognition (formants)

pitch tracking

-> retrieval of control parameters

+ some examples

vowel: wah-wahctor: control a guitar by saying wah-wah

pitch: singing trumpet: singing trumpet/bass control

advanced: beat-boxing

++ free speech software

+ FreeTTS

http://freetts.sourceforge.net/

pure Java TTS engine, diphone & concatenate

languages: EN

platforms: any Java2 v1.4 platform

+ flite

http://fife.speech.cs.cmu.edu/flite/

Flite (festival-lite) is a small, fast run-time synthesis engine primarily designed for small embedded machines and/or large servers.

platforms: runs on Desktop, PDAs, iPod Linux

+ Festival

http://www.cstr.ed.ac.uk/projects/festival/

command line tool, C++ API

languages: EN, ES, Welsh

platforms: win32, unix

++ speech toolkits

+ CMU Sphinx

http://cmusphinx.sourceforge.net/

open-source speech recognition package

not yes complete: for research & development purposes

pure Java version in preparation

+ HTK http://htk.eng.cam.ac.uk/

hidden markov model based speech recognition research package

source code available for research under NDA

+ mbrola http://tcts.fpms.ac.be/synthesis/

diphone TTS engine, european research project

languages: more than 25, construct DB for new language

platforms: any, binary only, free for non commercial use

++ voice interface programming

+ programming interfaces for speech applications

* provide access to existing speech recognition/synthesis engines

* can allow development of applications independently of the actual speech-engine vendor

* can increase platform portability of speech applications

++ voice programming: Java Speech API

+ platform independent cross-vendor API

http://java.sun.com/products/java-media/speech/

current version 1.0, published in 1998 by Sun

standard java extension: javax.speech.*

implementations include mostly TTS engines only

+ allows cross-platform java speech applications

independent of used speech engines

engines can be either native or implemented in Java

+ architecture

- provides synthesizer & recognizer objects
- event driven model
- mark-up languages for rule-grammars and text-formatting

++ mark-up languages: VoiceXML

+ XML based language for voice-browser apps

VoiceXML is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of web-based development and content delivery to interactive voice response applications.

+ client-server concept

- server delivers & processes XML content
- clients render & display XML contents

+ W3C standard: VoiceXML 2.0 draft

http://www.w3.org/TR/voicexml20/

VoiceXML 1.0 standard was published in 2000 by the

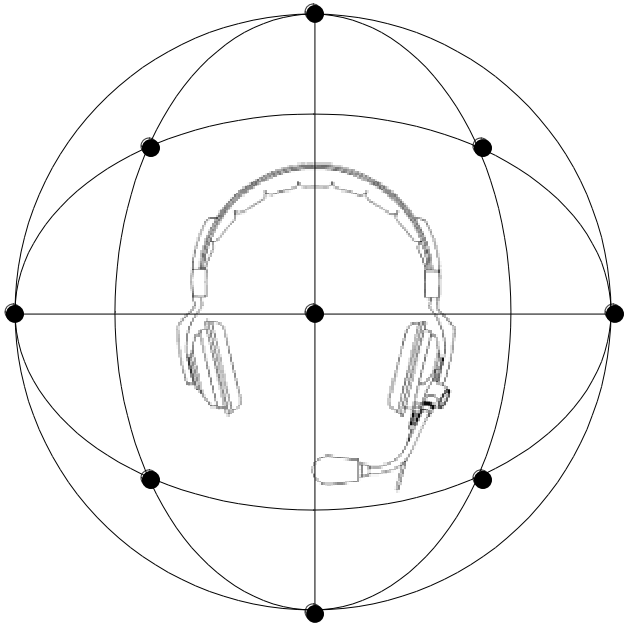
+ VoiceXML consortium

association of major industry leaders: http://www.voicexml.org/

IBM, AT&T, Lucent Bell Labs, Motorola, etc.

Multimodal Interfaces

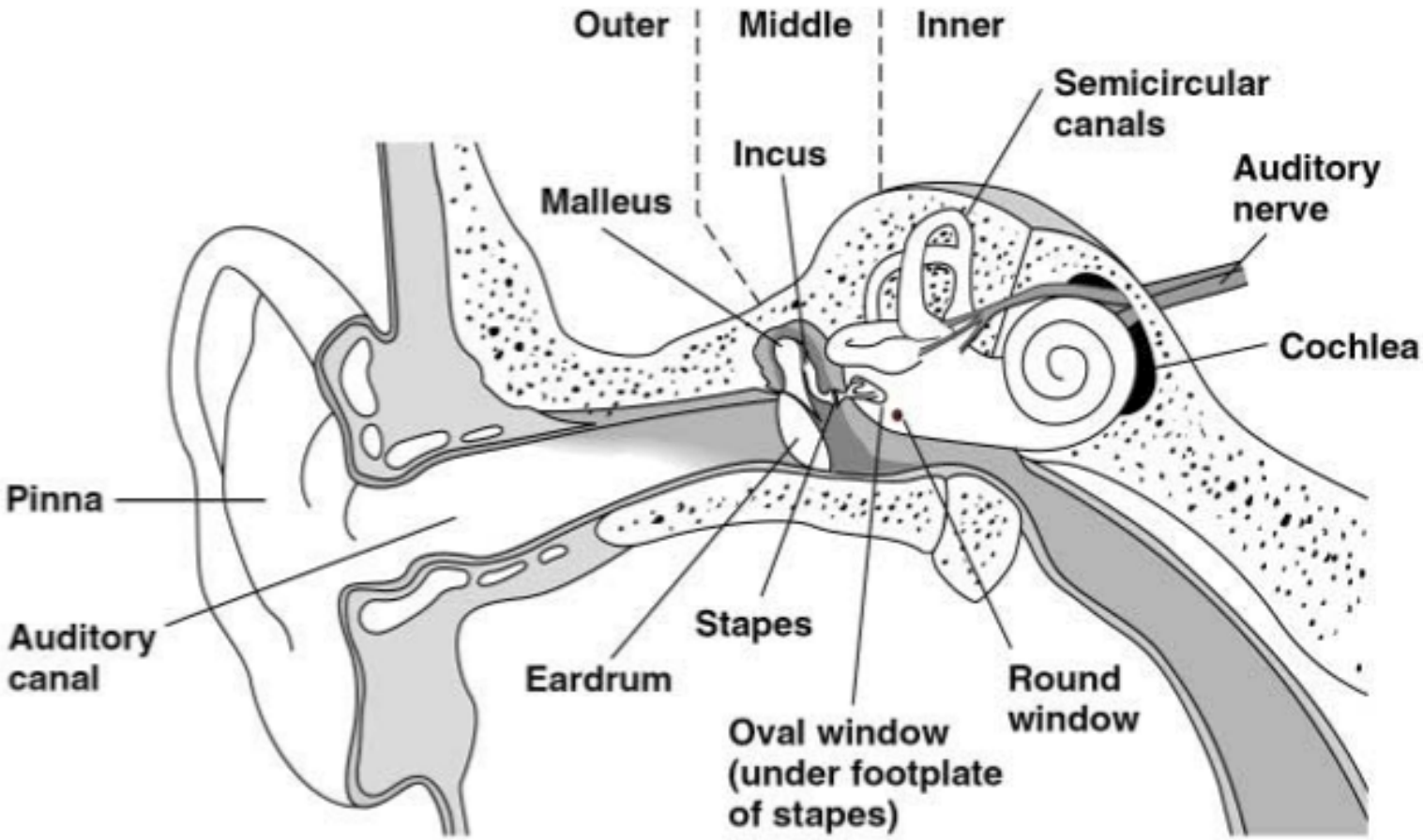
lecture 03c: auditory perception



Univ.Prof. Martin Kaltenbrunner

Interface Culture Lab, Kunstuniversität Linz

++ the human ear



++ hearing process

+ outer ear

pinna focuses sound, spatial hearing

auditory canal, resonance tube

2.000Hz - 5.000Hz (human speech)

+ middle ear

mechanical amplification, 1.3 gain

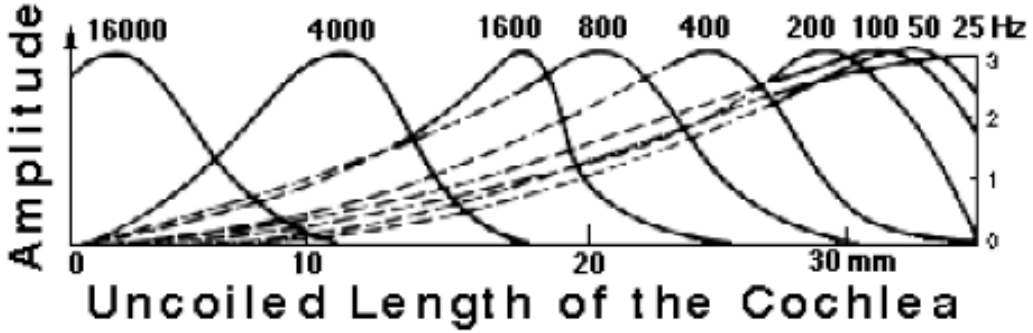
hammer -> anvil -> stirrup carry vibration to cochlea

+ inner ear

cochlea filled with perilymph fluid, unrolled 35mm

in its centre vibrates basilar membrane

each region resonates at different frequency, hair cells transmit signal



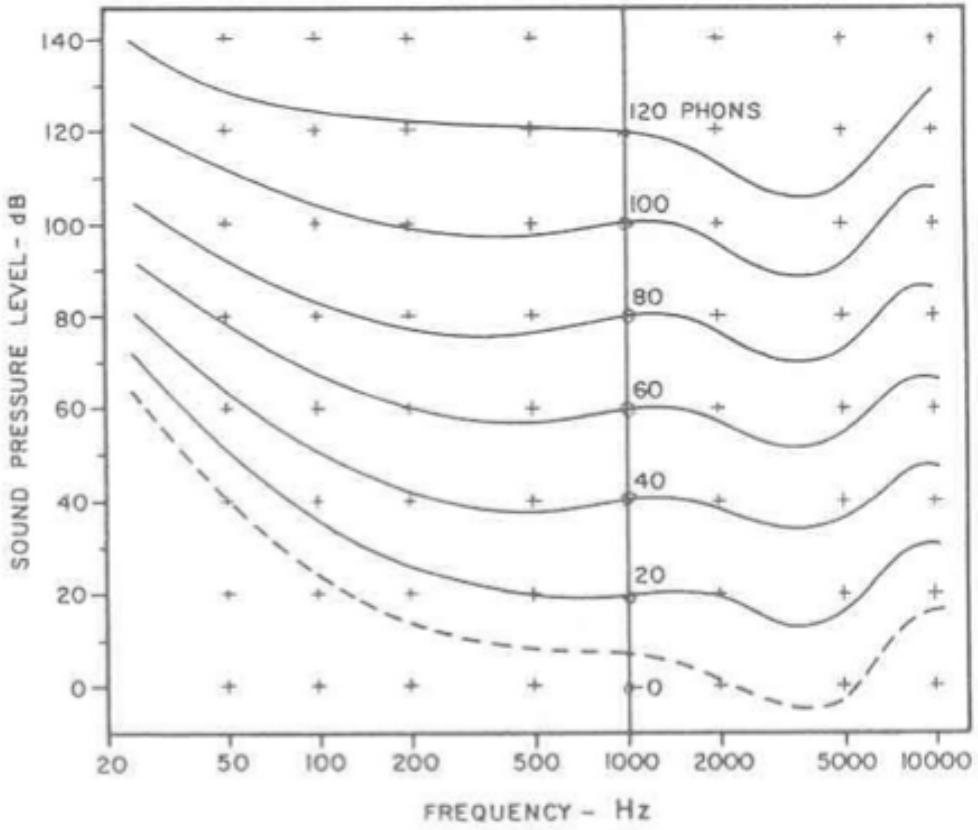
++ loudness perception

+ loudness

the human ear responds ~ logarithmically to loudness

decibel (dB) ~ 20 log air pressure

+ loudness at different frequencies



fortissimo	fff
	ff
	f
	p
	pp
pianissimo	ppp

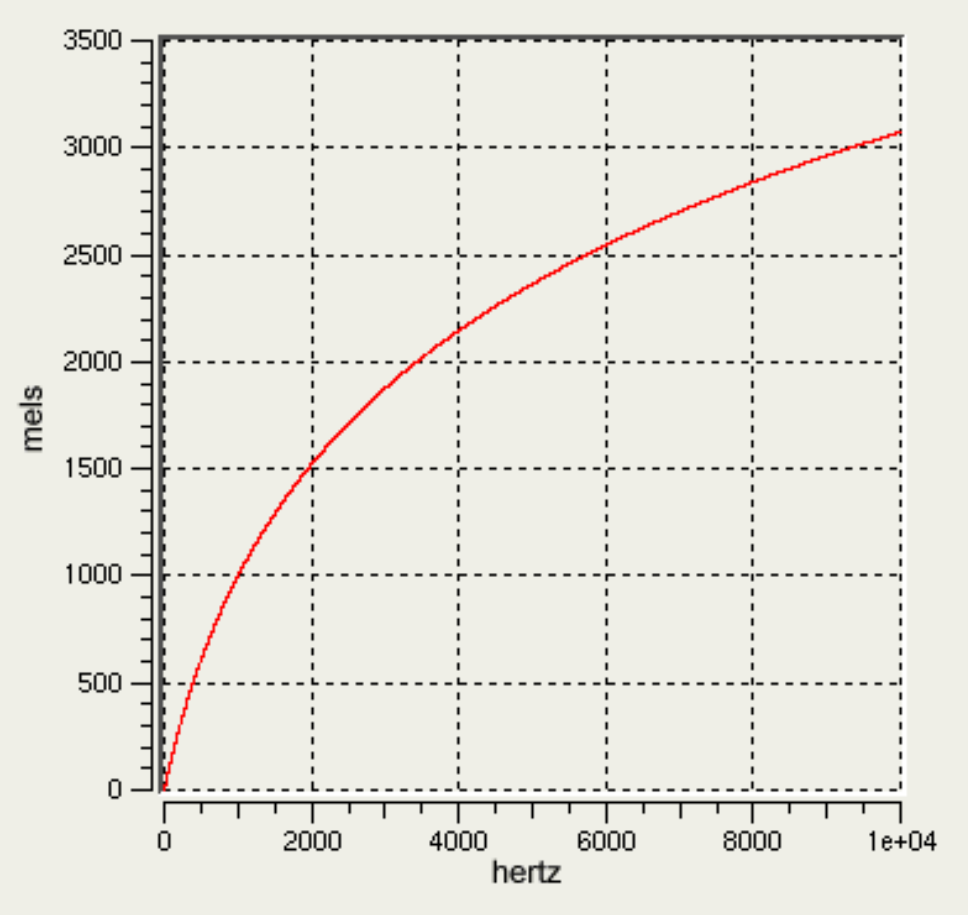
10^{-2} W/m^2	L (dB) = 100
10^{-3}	90
10^{-4}	80
10^{-6}	60
10^{-7}	50
10^{-8}	40

++ pitch perception

+ mel scale

A perceptual scale of pitches judged by listeners to be equal in distance one from another. The reference point between this scale and normal frequency measurement is defined by equating a 1000 Hz tone with a pitch of 1000 mels.

Above about 500 Hz,
Larger intervals are judged by listeners to produce equal pitch increments.
As a result, four octaves on the hertz scale are judged to comprise about two octaves on the mel scale.



++ spatial audio: stereo

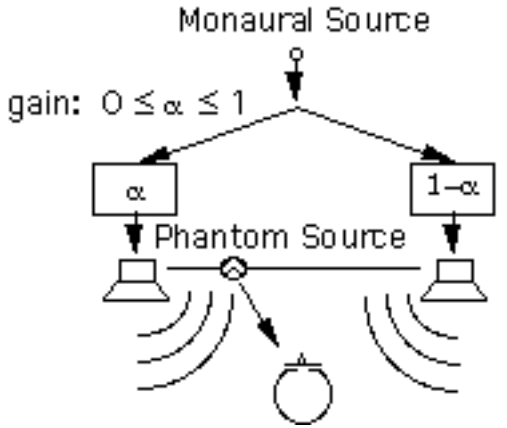
+ two-speaker systems

sound to the left or right speaker

cross-fading sound from left to right

creates “phantom source” in the centre

but always exactly on the line



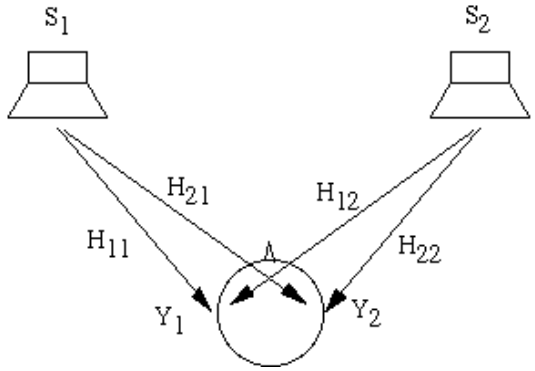
+ cross-talk cancellation

problem: if not using headphones

signal Y1 on left ear is mix of S1*H11 & S2*H12

signal Y2 on right ear is mix of S2*H21 & S1*H22

solution: inverting this process



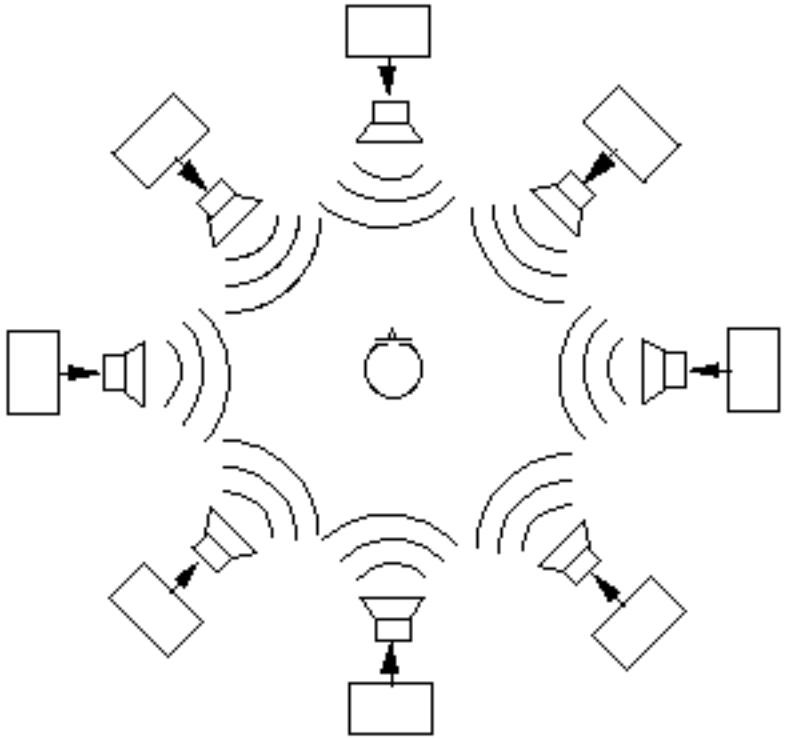
$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \qquad \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$$

++ spatial audio: multi-channel

+ **array of spatially arranged speakers**

+ **subwoofer for non-directional low-frequencies**

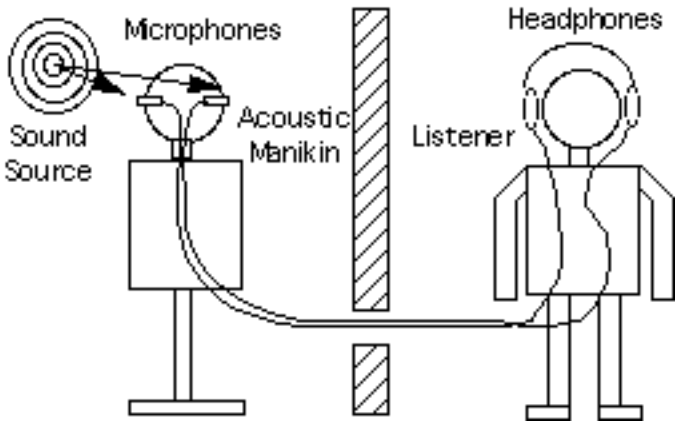
  in reverberant environments
  high frequency content
  is more important for localisation
  (franssen effect)



++ spatial audio: binaural recordings

+ two channels

sufficient when using head-phones



+ listener's or artificial head

recording with two microphones places

in a dummy head´s ears produce realistic sound

+ disadvantages

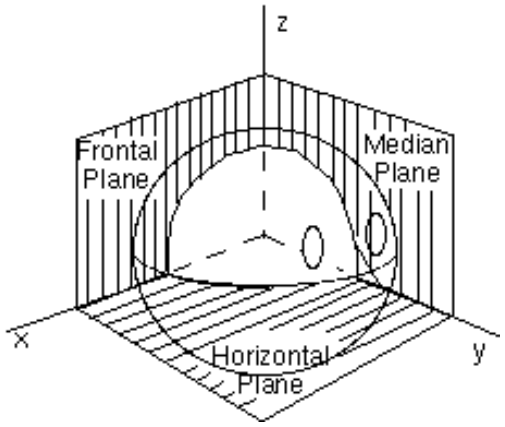
- require headphones
- if the listener moves the sound source moves as well
- not interactive (recordings)

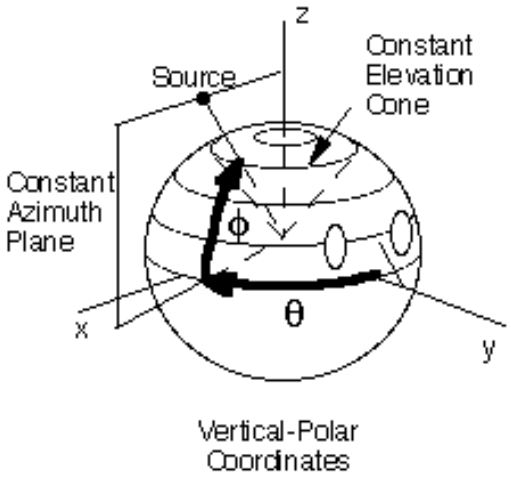
+ synthesis

using HRTFs with head-tracking

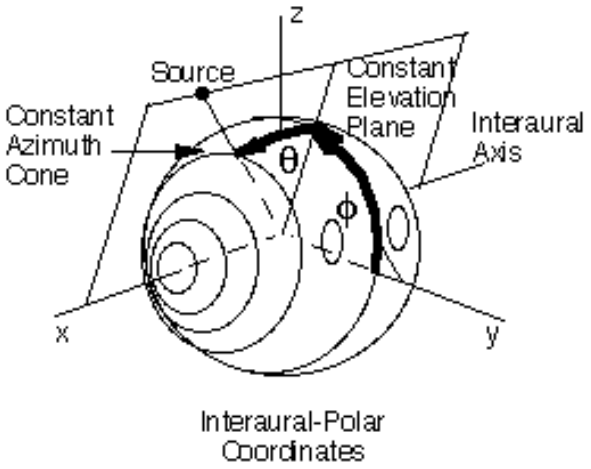
adding room reflections and reverberation

++ spatial hearing: coordinate system





Vertical-Polar Coordinates



Interaural-Polar Coordinates

+ cartesian coordinates

X-axis through the ear, Y-axis through the nose, Z-axis vertically

horizontal plane, frontal plane, median plane

+ vertical-polar coordinates

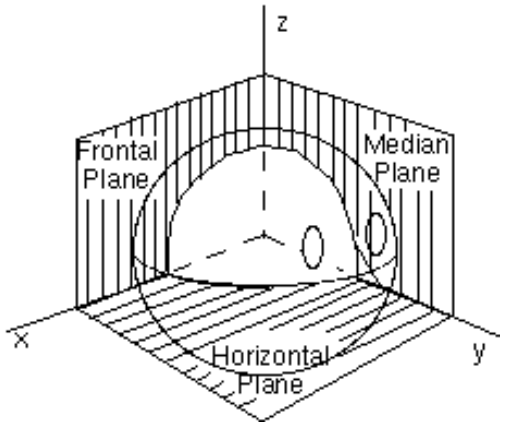
azimuth in horizontal plane +/- 180 deg, elevation +/- 90 deg

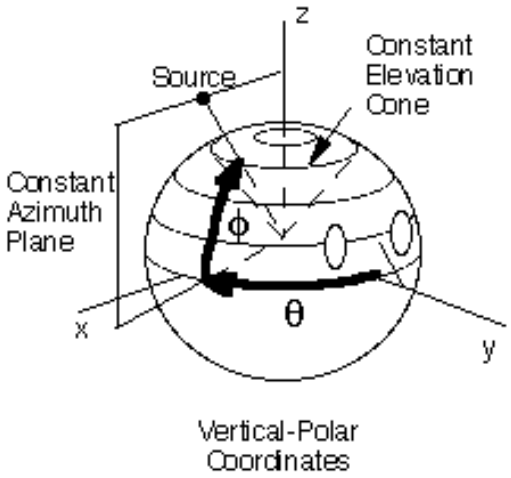
clearer for definition of front/back in horizontal plane

+ interaural-polar coordinates

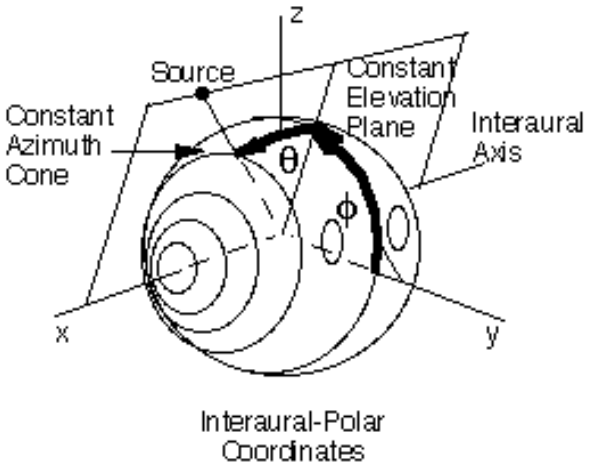
azimuth +/- 90 deg, elevation +/- 180 deg

++ spatial hearing: coordinate system





Vertical-Polar Coordinates



Interaural-Polar Coordinates

+ cartesian coordinates

X-axis through the ear, Y-axis through the nose, Z-axis vertically

horizontal plane, frontal plane, median plane

+ vertical-polar coordinates

azimuth in horizontal plane +/- 180 deg, elevation +/- 90 deg

clearer for definition of front/back in horizontal plane

+ interaural-polar coordinates

azimuth +/- 90 deg, elevation +/- 180 deg

++ spatial hearing: azimuth cues

+ interaural time difference ITD

sound travels at 343 m/s

--> sound takes around 0.7ms longer to other ear (on average head)

+ interaural intensity difference IID

head shadow effect: head shades sound intensity

at high frequencies > 1.5kHz up to 20dB level difference

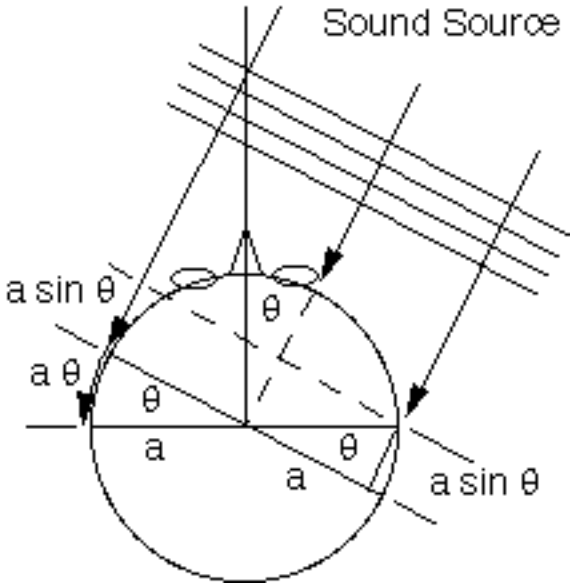
+ duplex theory

low frequencies:

- little IID effect
- ITD compensates

high frequencies:

- little ITD effect
- IID compensates



++ spatial hearing: elevation cues

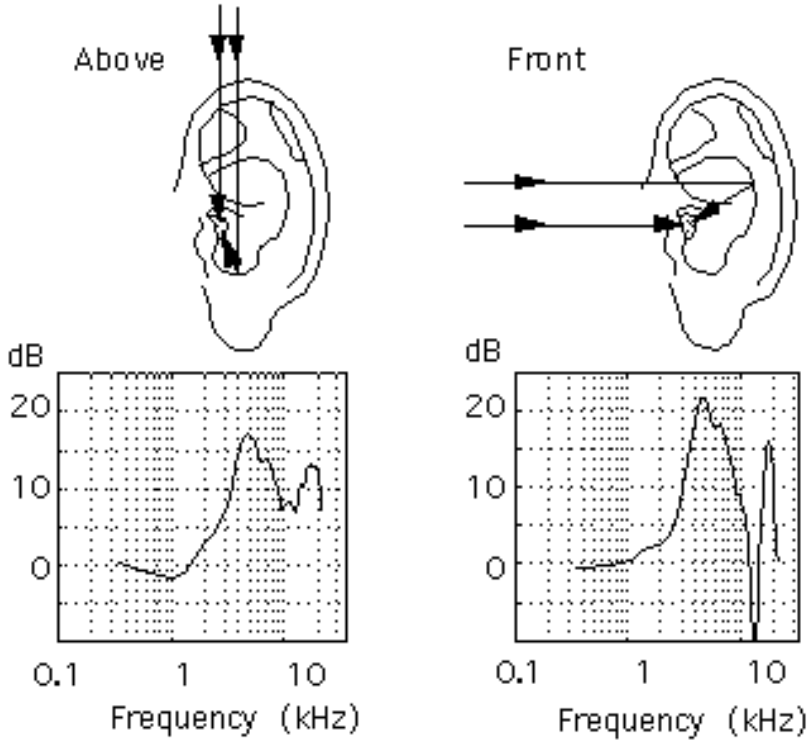
+ outer ear: pinna

the special shape of the pinna amplifies or lowers certain frequencies differently when coming from above, below, front, back
-> directionally dependent frequency response

+ “pinna notch”

phase cancellation at ~10kHz
direct sound/ reflected sound

different “notch” for the various source directions



++ spatial hearing: range/distance cues

+ loudness

distant sound is lower in intensity than nearby sound (square of range)

problem: close low volume sound, loud distant sound?

+ motion parallax

moving the head produces greater azimuth changes for closer sources

approaching sounds also produce bigger IID (if closer than one meter)

+ direct sound/reverberation ratio

clear distinction of close and distant sources

-> more energy in direct sound

++ binaural synthesis: HRIR, HRTF

+ KEMAR head

artificial head, acoustic manikin

for binaural recordings

in anechoic chambers

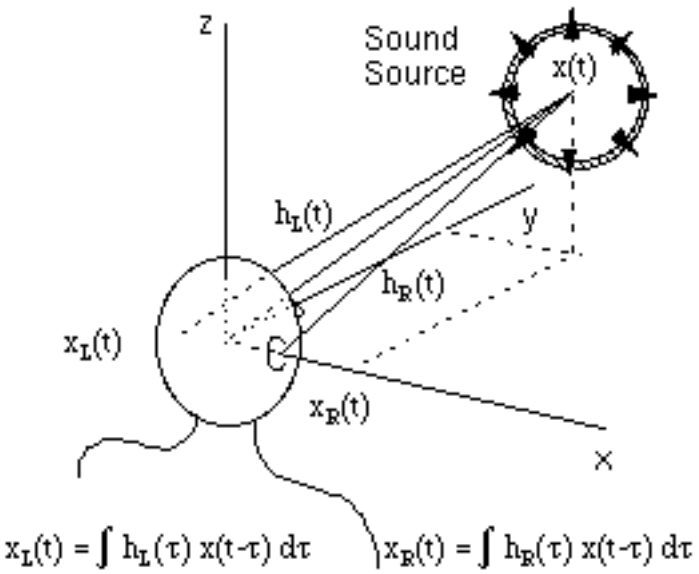
+ IR recordings

impulse responses of a

monaural sound source

at various spatial far field locations

for both ears



+ HRIR: head related impulse response

impulse response from a sound source in the ear drum

+ HRTR: head related transfer function

fourier transform of the HRIR

a function of azimuth, elevation (distance) and frequency

++ room models

+ room impulse response

consists of direct sound, early reflections and a reverberation tail

convolution with the Fourier transform creates realistic room image

+ calculation

each room impulse response hast to be recalculated for:

- different rooms
- different positions of the listener
- different sound source positions

+ simplification

this process is very costly, simplifications for spatial audio include:

- simple rectangular rooms
- few early reflections
- generalized reverberation tail

++ room models

+ early reflections

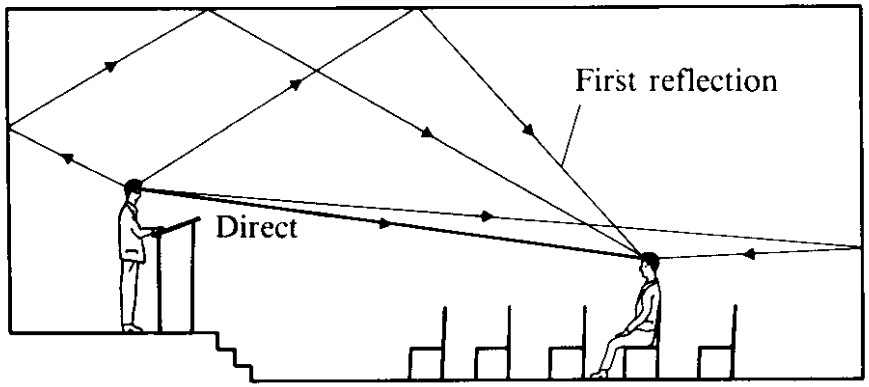
after the direct sound early reflections, which are reflected from the room´s walls reach the listener's ears after 50 milliseconds.

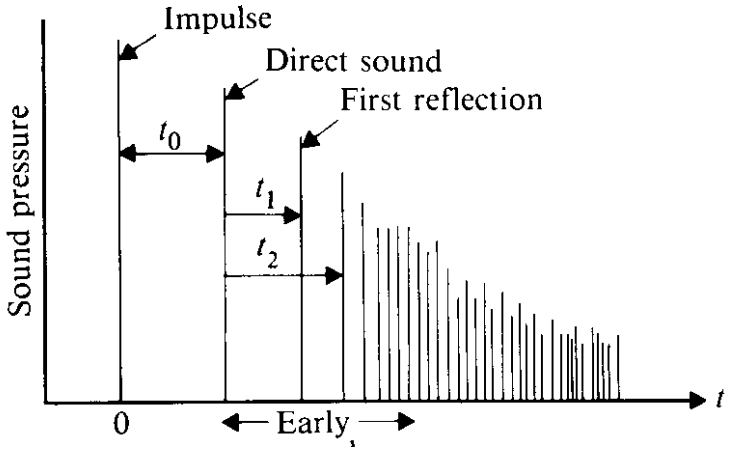
a simple model for early reflections is the placement of various "virtual" sound sources outside of the simulated room

precedence effect: law of the first wave front

+ reverberation tail

after 100 milliseconds the reflections are not calculated anymore because the rest of the signal is just diffuse sound.





++ spatial audio programming

+ OpenAL

an attempt to create an “OpenGL” standard for spatial audio

http://www.openal.org/

+ DirectSound

Microsoft´s audio part of the DirectX multimedia API support spatial audio. Various sound cards provide hardware 3D audio support.

+ 3D APIs

various 3D APIs such as Java3D or OpenInventor come with spatial sound functionality which is integrated into their scene graph models.

VRML has some 3D sound capabilities as well

+ others

fmod, multi-platform closed source library, http://www.fmod.org/