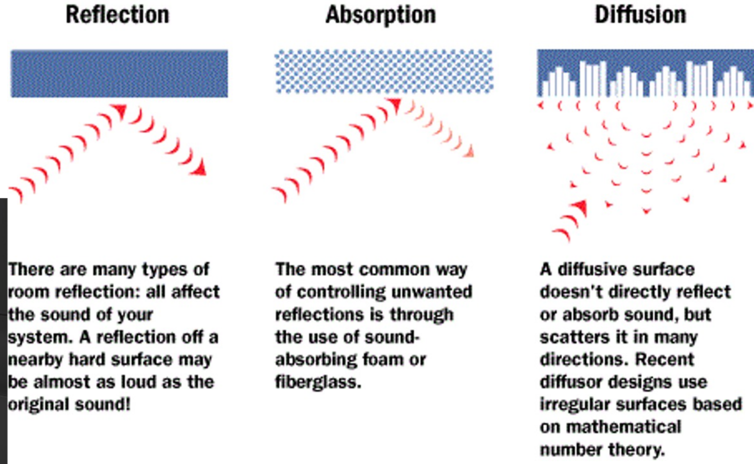
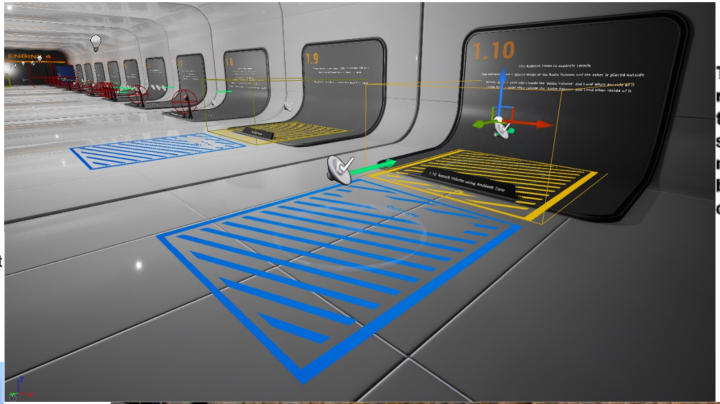
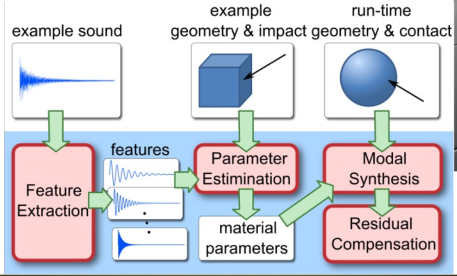
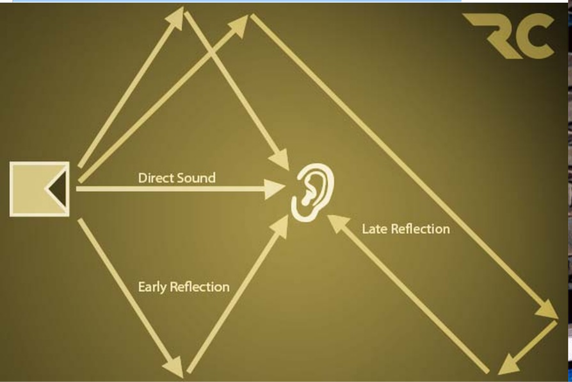
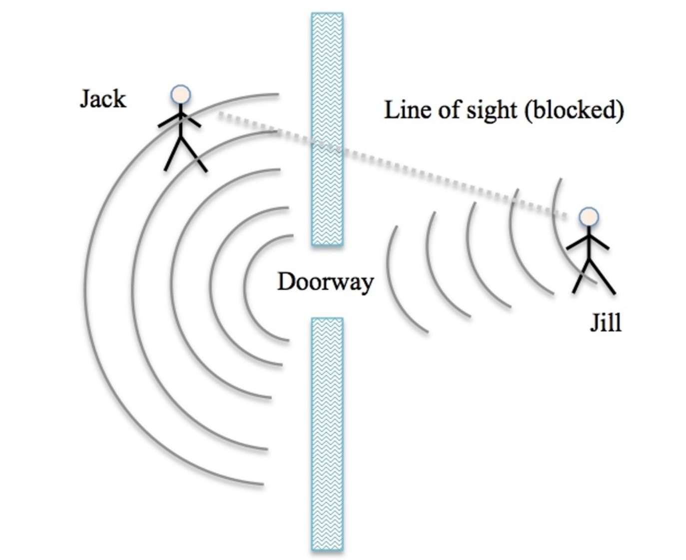
Introduction to 3D Audio







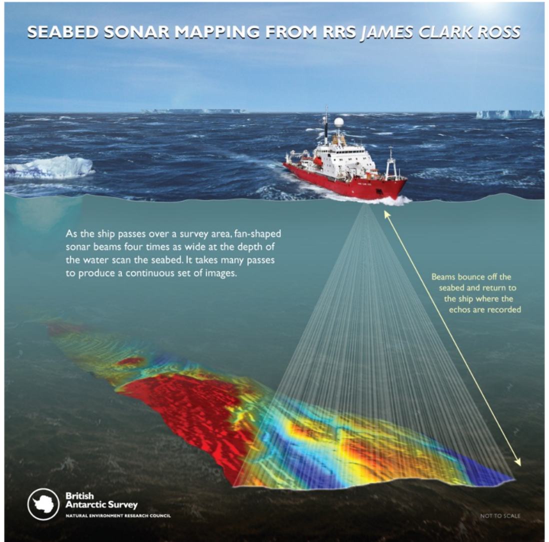


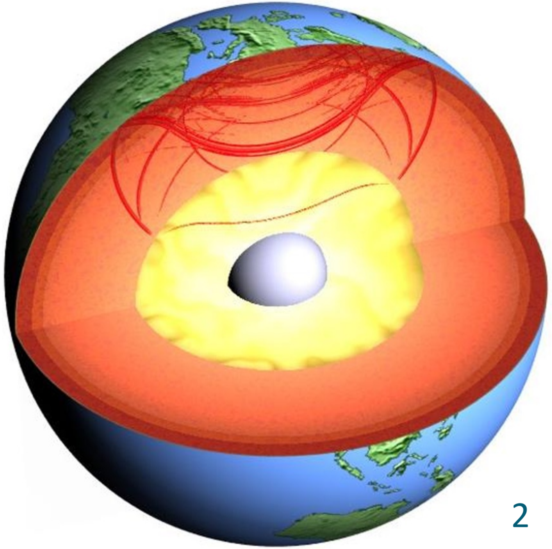


What is Sound?

Elastic wave propagation in some medium

- Earth Science (Seismic waves)
- Ocean Acoustics (Sonar)
- UltraSound (Human Tissue)

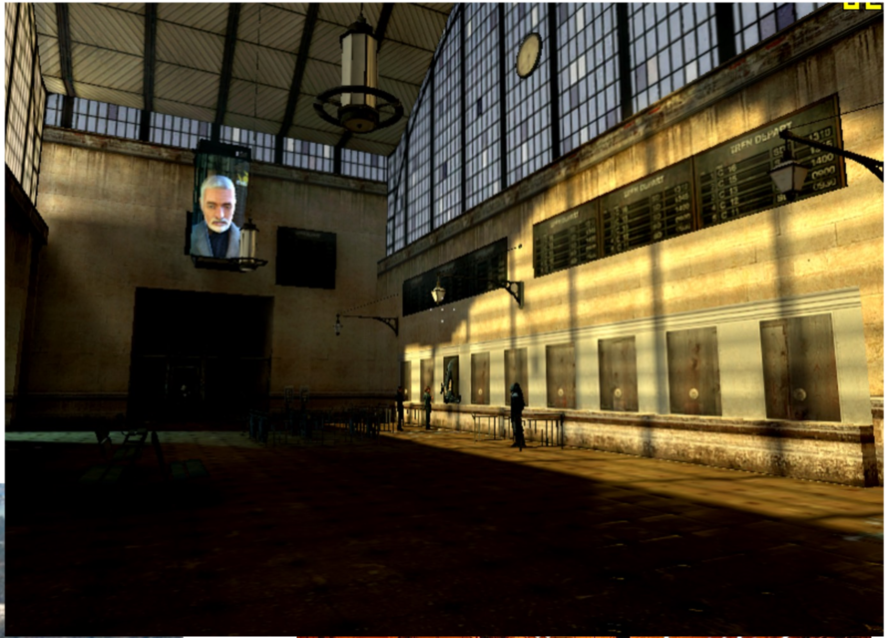




What is Sound?

Elastic wave propagation in some medium

- Structural Mechanics
- Architectural Acoustics
- Games



Immersive Environments

- Imitation of reality in a computer-generated world
- Graphics hardware and techniques have evolved

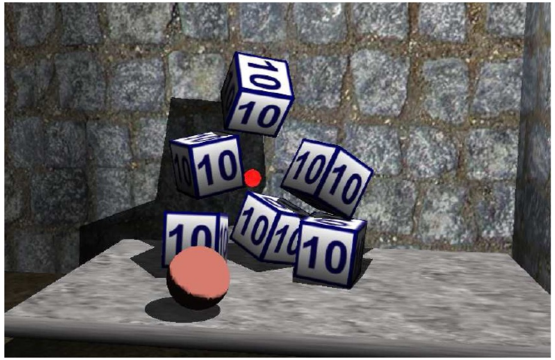
How can it be done?

- Foley artists manually make and record the sound from the real-world interaction

Lucasfilm Foley Artist

How about Computer Simulation?

Physical simulation drives visual simulation



- Sound rendering can also be *automatically* generated via 3D physical interaction

Immersive Audio

- Goal of immersive audio: Realistic *Auralization*
- Reality: Hearing and Sight work together
- Hearing provides additional information that complements Sight
- A natural approach: Physically-based Sound Simulation

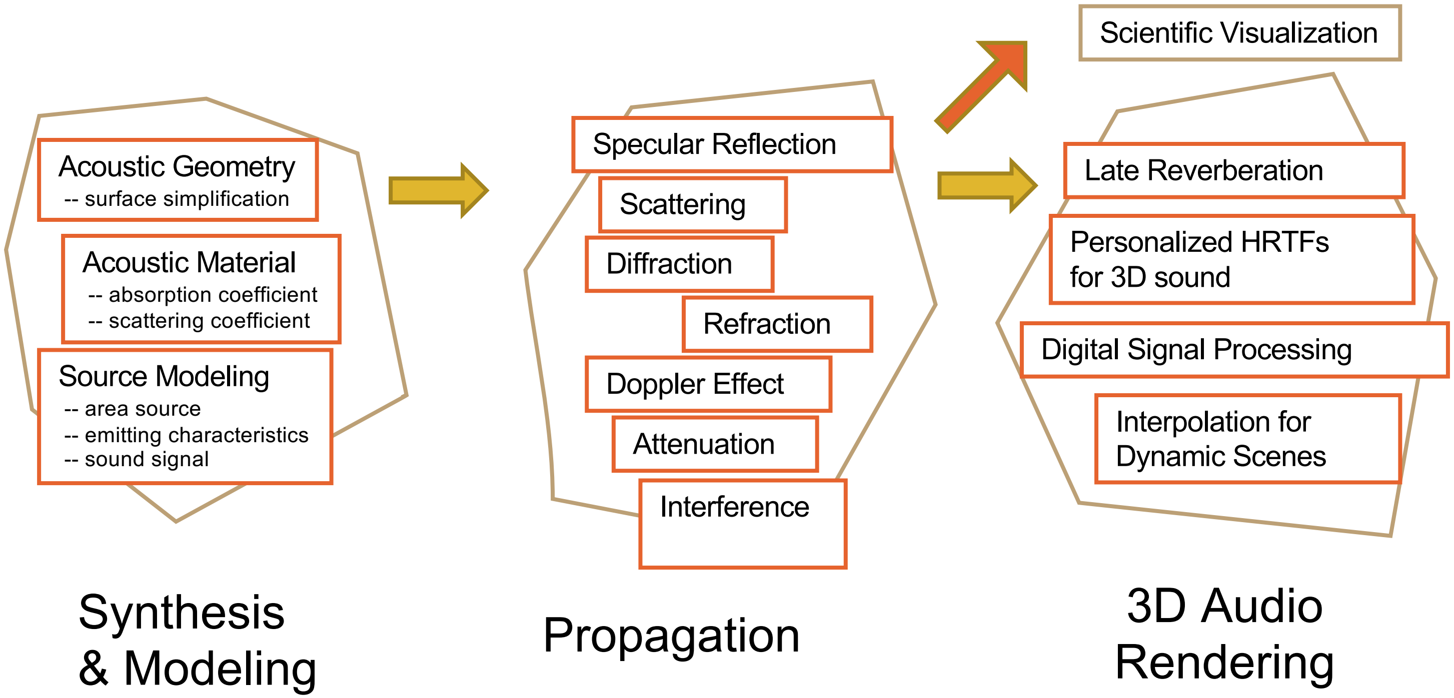
Importance

- **Localization**: User should know where sound source is located
- **Avoid confusion**: Position should not be ambiguous unless intended to be
  - Front-back confusion: Common in VR; user doesn't know if source is in front or back of them (often described as sound coming from inside their head)
- **Maximize immersion**: Good audio increases immersion

Example of non-VR great 3D audio (ASMR-like):
https://www.youtube.com/watch?v=IUDTlvagjJA

Audio makes a difference!

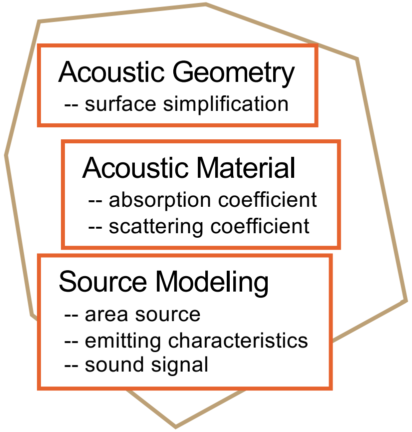
Sound Rendering: An Overview



Synthesis & Modeling

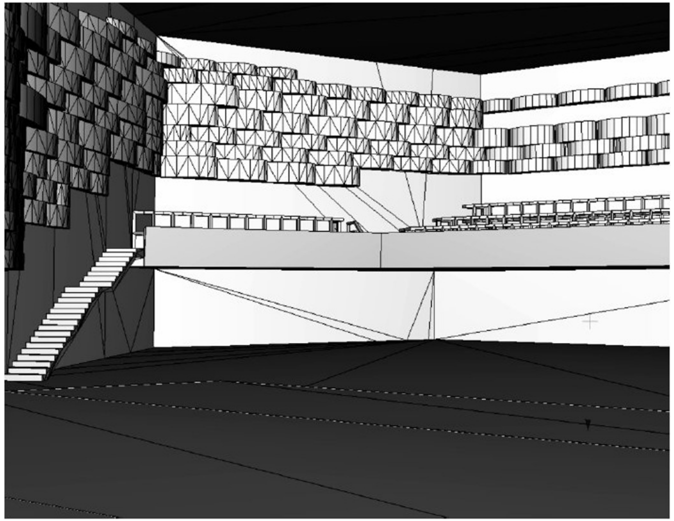
Acoustic vs. Graphics

- Low geometric detail vs. High geometric detail

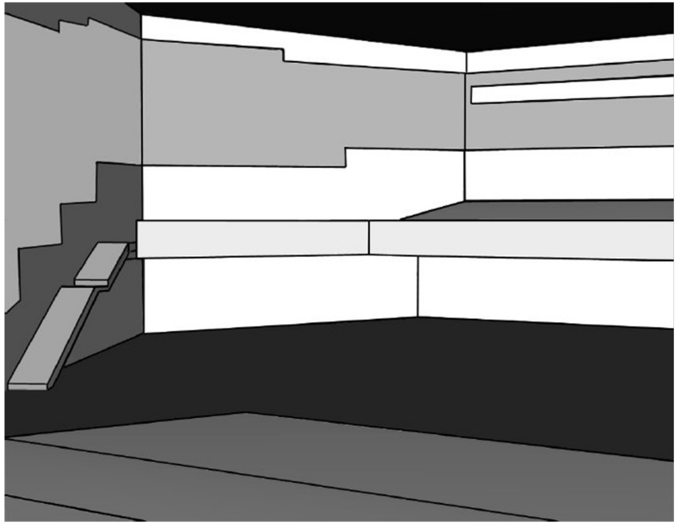


Synthesis & Modeling

Modeling Acoustic Geometry [Vorländer,2007]

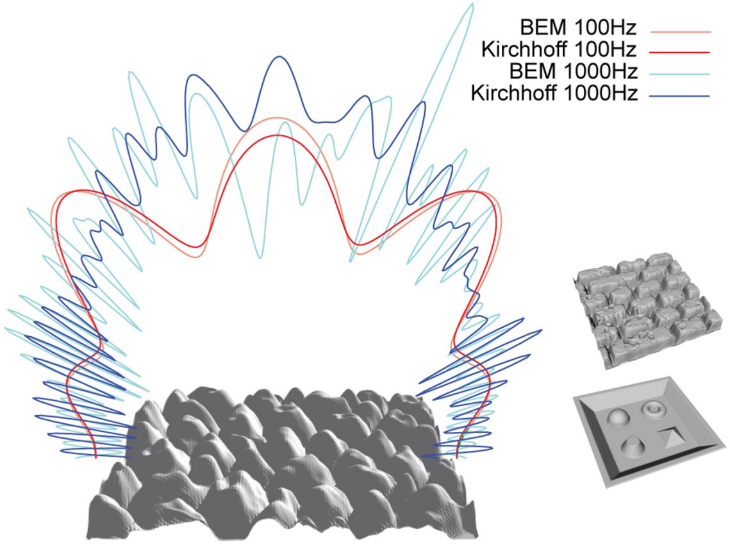


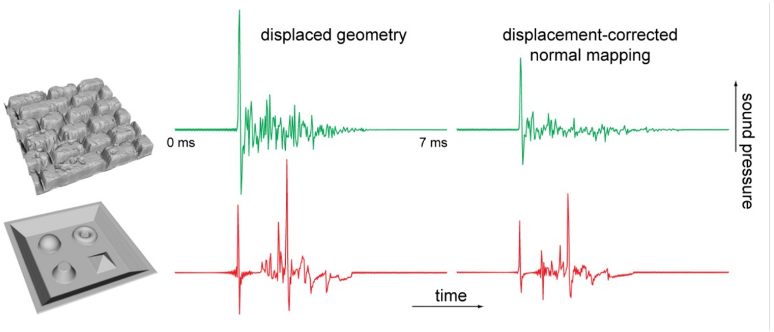
Visual Geometry



Acoustic Geometry

Modeling Sound Material

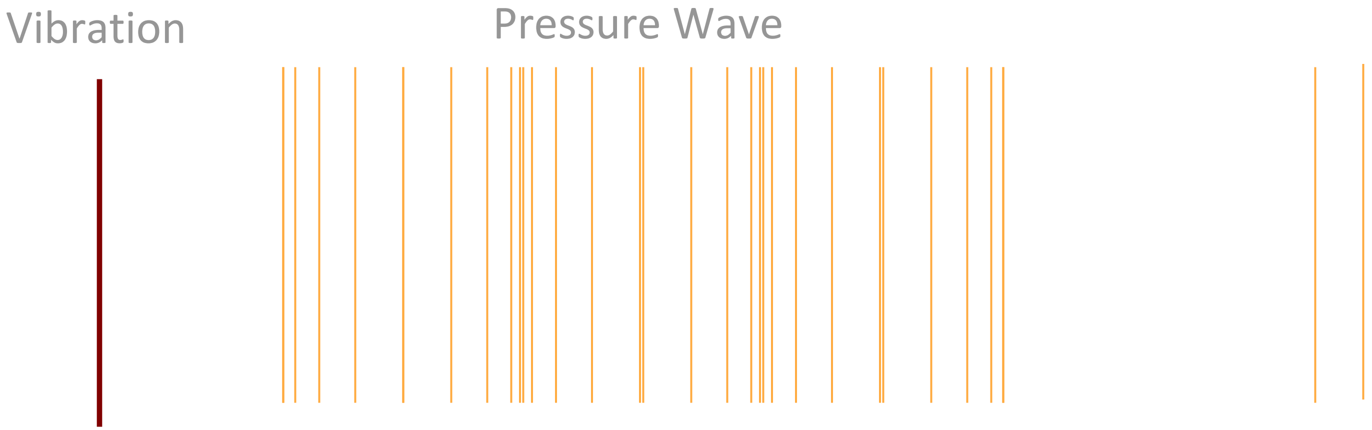




[Embrechts,2001] [Christensen,2005] [Tsingos,2007]

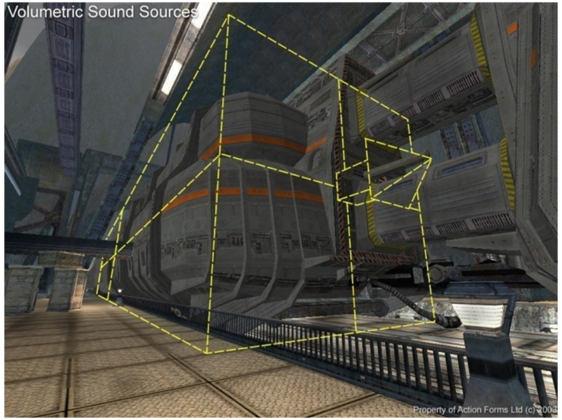
Sound Synthesis

- Collisions lead to surface vibrations
- Vibrations create pressure waves in air
- For small amplitudes, linear phenomenon

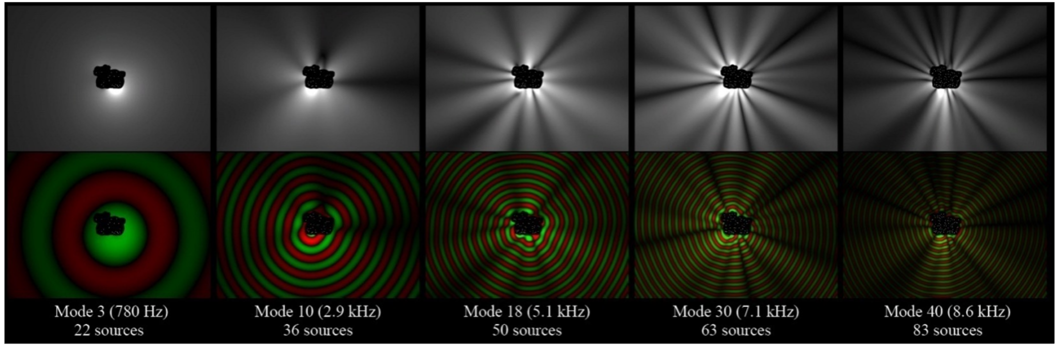


Modeling Sound Source

Volumetric Sound Source



Directional Sound Source

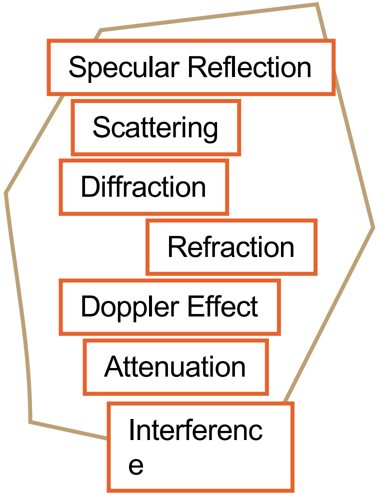


Complex Vibration Source

Propagation

Acoustic vs. Graphics

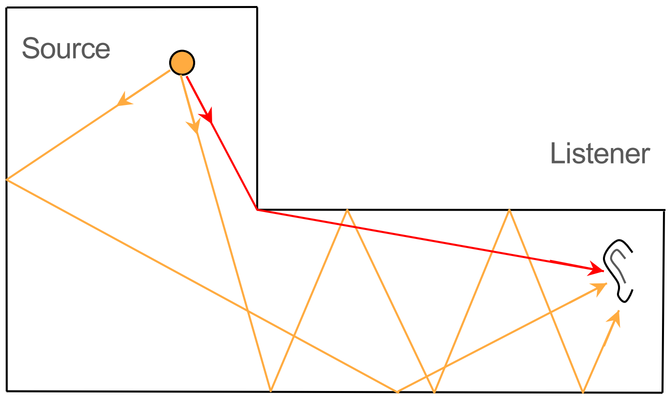
- 343 m/s vs. 300,000,000 m/s
- 20 to 20K Hz vs. RGB
- 17m to 17cm vs. 700 to 400 nm



Propagation

Sound Propagation

- Sound waves propagate from source, complex interactions with boundary
- Diffraction, high-order reflection, scattering



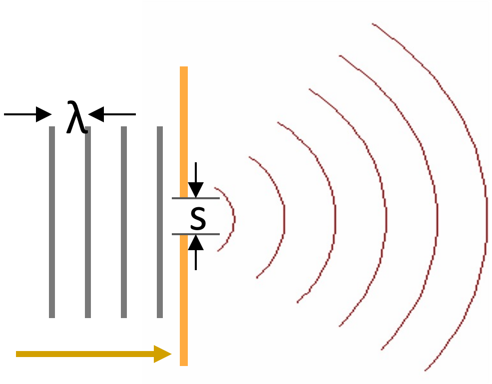
Diffraction

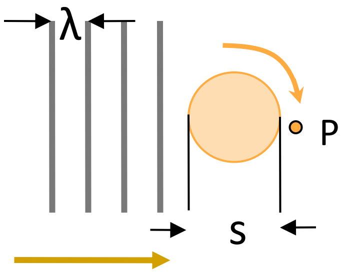
- Waves tend to bend around obstacles when

  $\lambda > s$

- P will have appreciable reception only if there is a good amount of diffraction

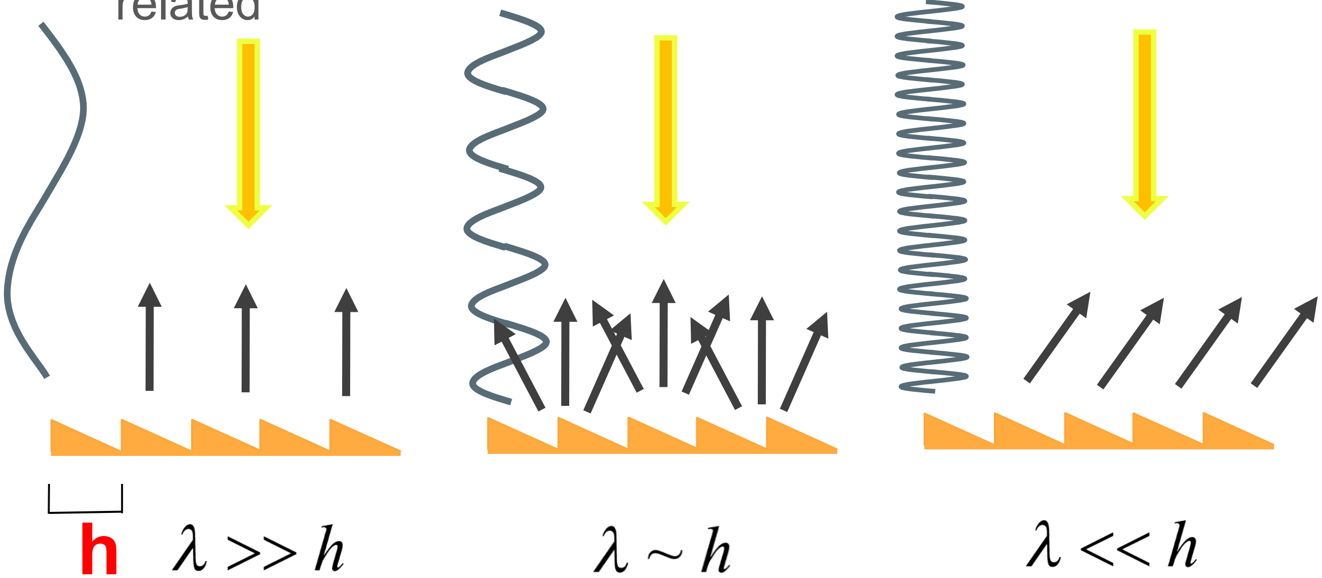
- Low-frequencies tend to "bend" more than higher frequencies





Scattering

- For sound, scattering and diffraction are closely related



Physical Properties: Sound and Light

Physical Property	Light	Sound
Speed of propagation	~ 300,000,000 m/s	~ 340 m/s
Observable Wavelength	380 – 750 nanometers	17 mm – 17 m
Observable Frequency	400 – 790 TeraHertz	20 – 20,000 Hertz

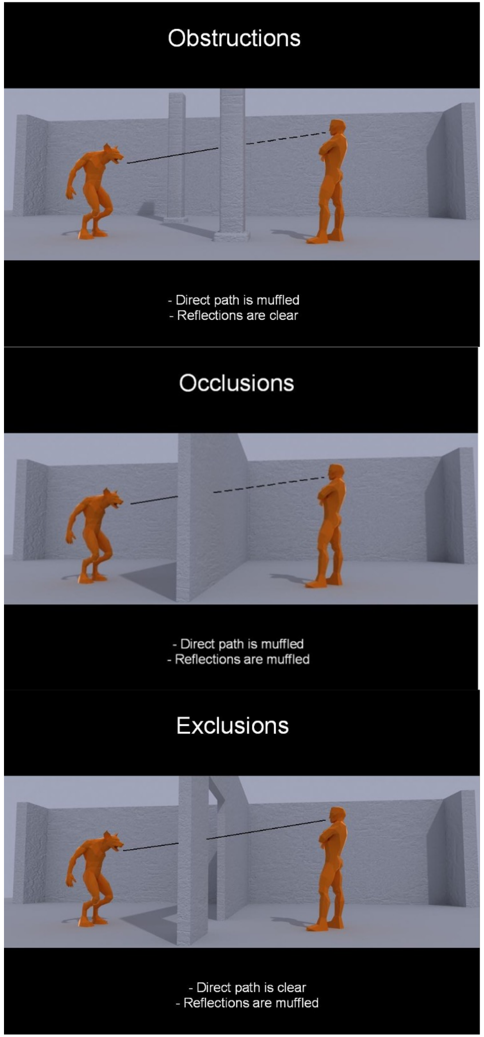
Transient phenomena perceivable

Diffraction is important

High update rate

Sound Propagation in Games

- Strict time budget for audio simulations
- Games are dynamic
  - Moving sound sources
  - Moving listeners
  - Moving scene geometry
- Trade-off speed with the accuracy of the simulation
- Static environment effects (assigned to regions in the scene)



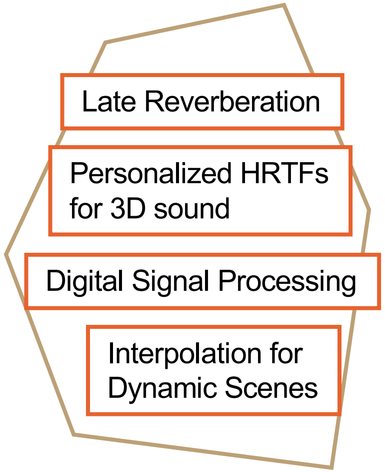
Sound Simulation: Sub-problems

- Sound Synthesis
    - How sound is produced locally due to an object's vibration
- Sound Propagation
    - How sound travels in a scene, up to the listener's ears
- Sound Perception (Psycho-acoustics)
    - How sound is processed by the human auditory system
- Interactive approaches: Pre-processing and Runtime

3D Audio Rendering

Acoustic vs. Graphics

- Compute intensive DSP vs. addition of colors
- 44.1 KHz vs. 30 Hz
- Psychoacoustics vs. Visual psychophysics

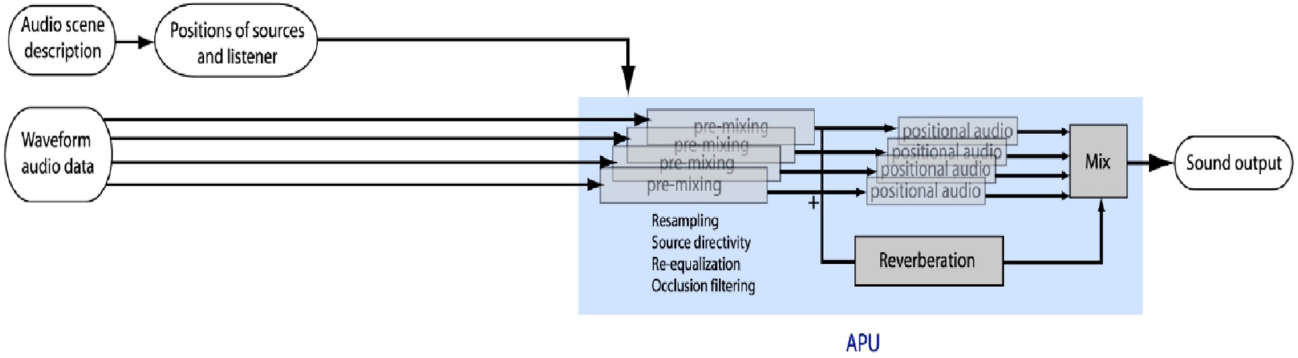


3D Audio Rendering

3D Audio Rendering

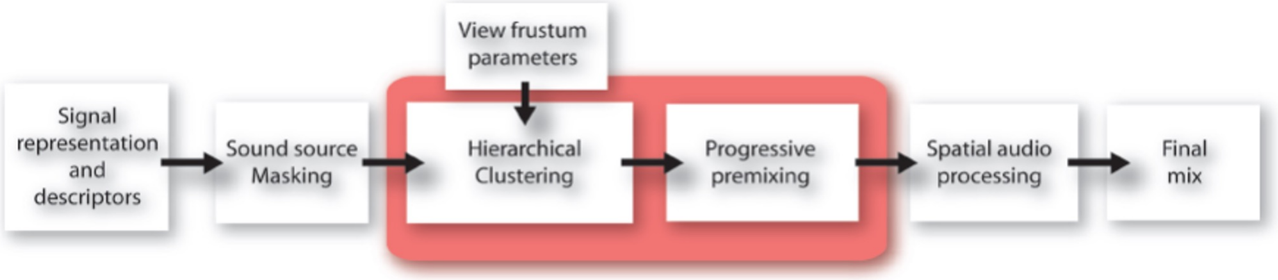
- Main Components
  - 3D Audio and HRTF
  - Artifact free rendering for dynamic scenes

Traditional pipeline

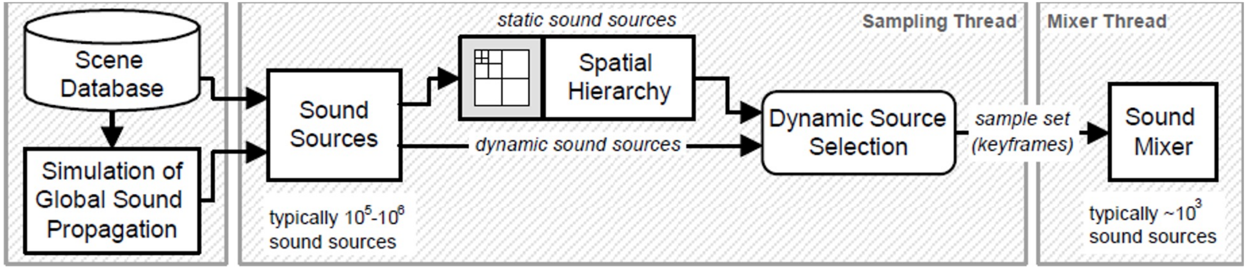


3D Audio Rendering

- Perceptual Audio Rendering [Moeck,2007]



- Multi-Resolution Sound Rendering [Wand,2004]

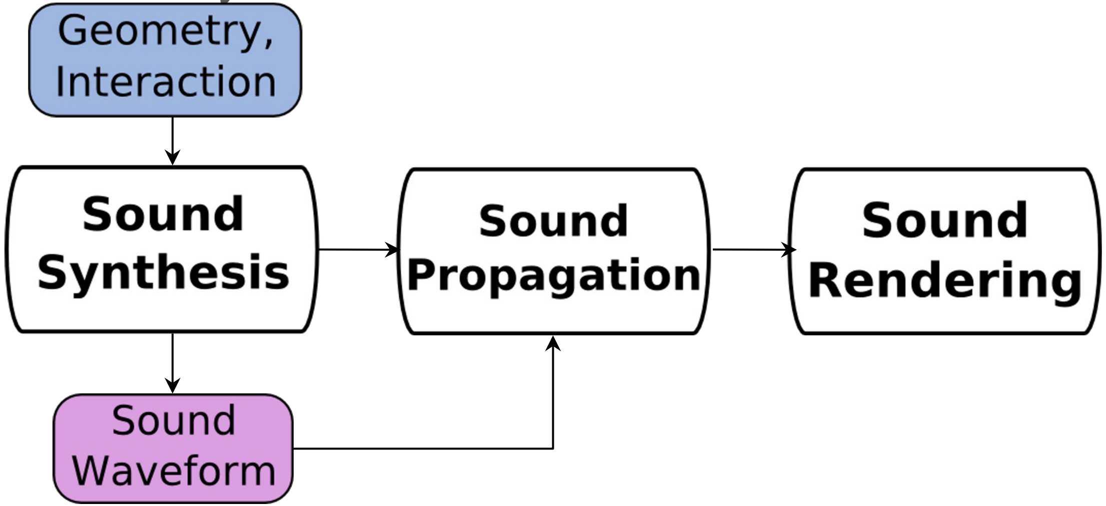


Overview of Sound Simulation

- The complete pipeline for sound simulation
  - Sound Synthesis
  - Sound Propagation
  - Sound Rendering

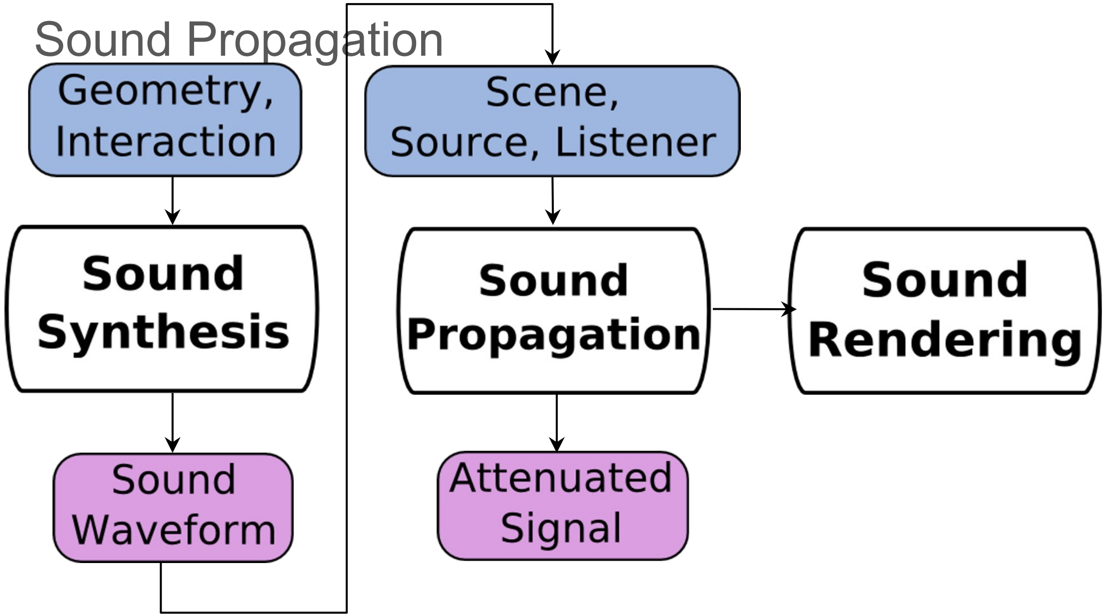
Overview of Sound Simulation

- Sound Synthesis



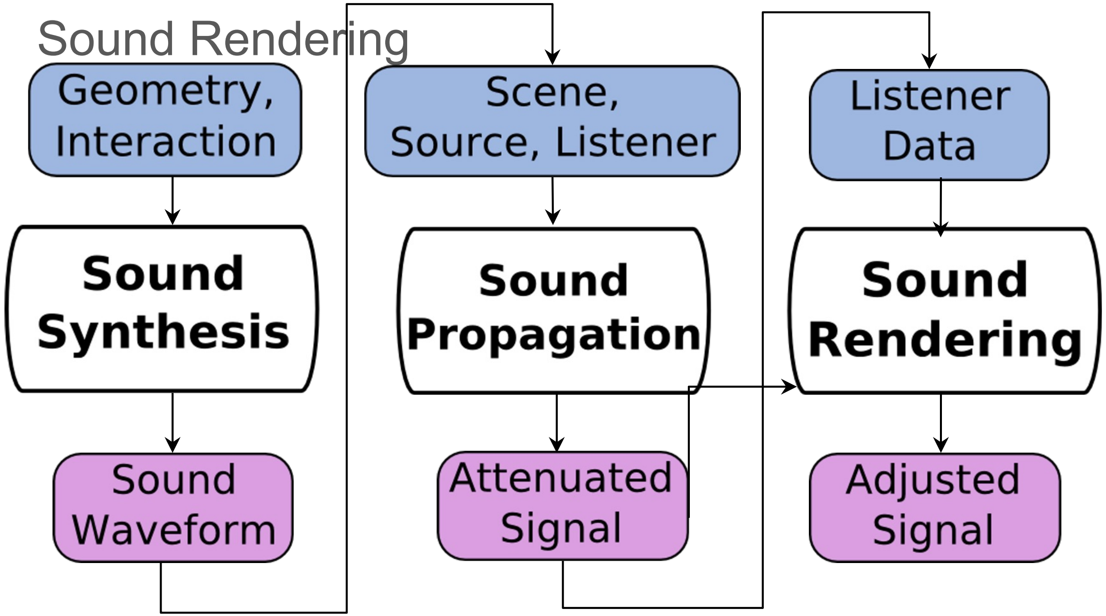
Overview of Sound Simulation

- Sound Propagation



Overview of Sound Simulation

- Sound Rendering

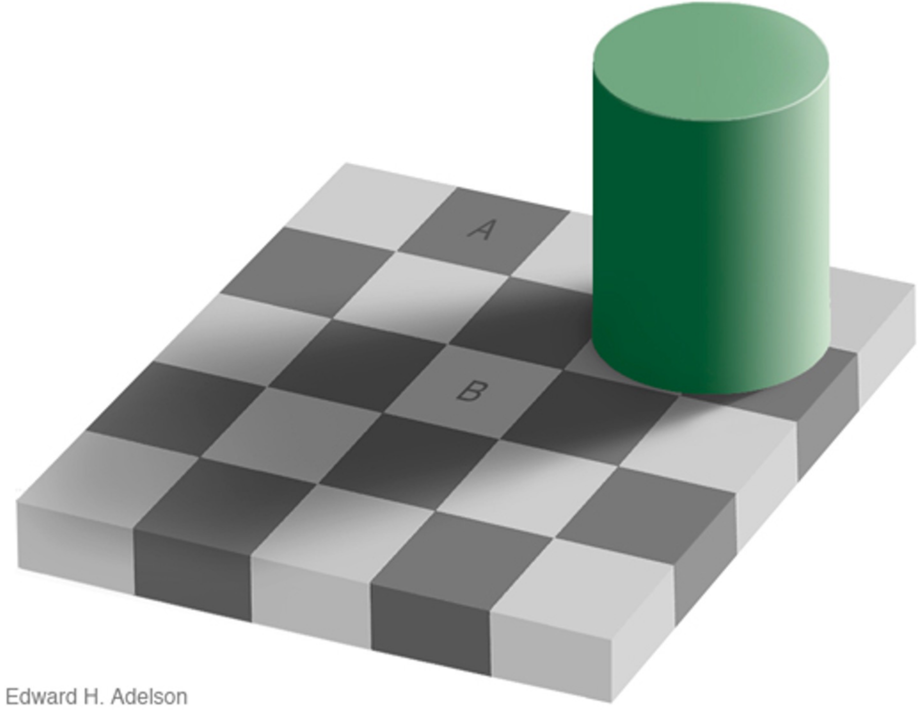


Themes

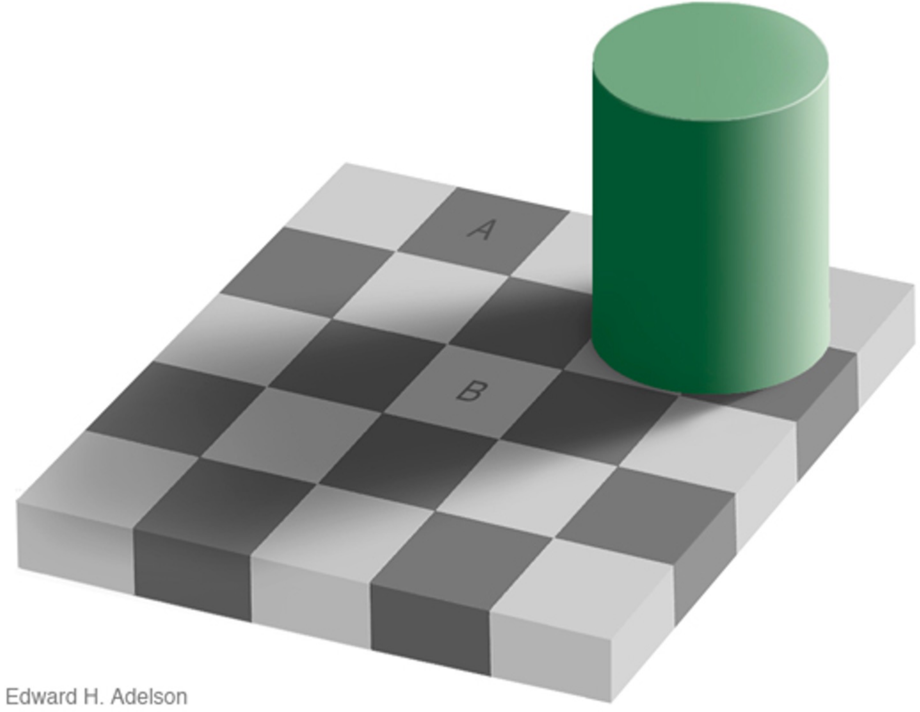
- Exploiting analytical solutions using Modal Analysis to accelerate numerical simulation and reducing runtime computation

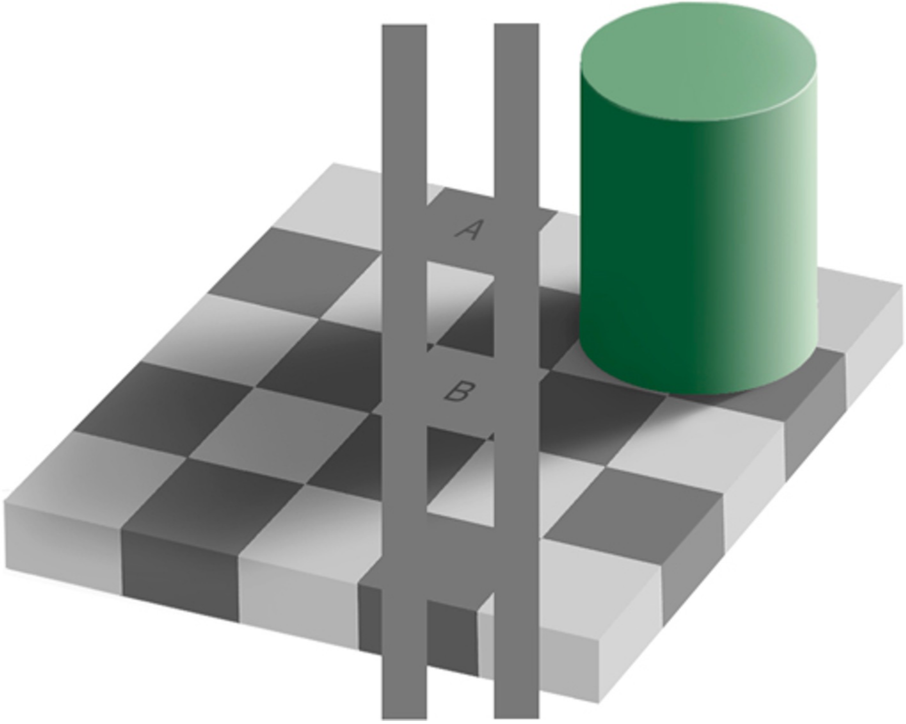
- Capture only perceptually important auditory cues to perform real-time sound synthesis and acoustic propagation on complex 3D scenes

Color Illusion



Color Illusion



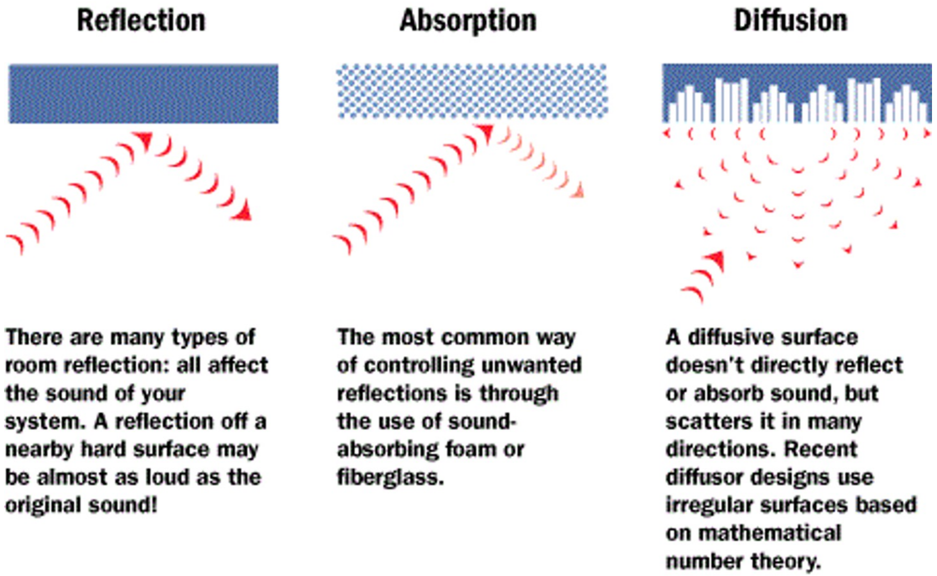


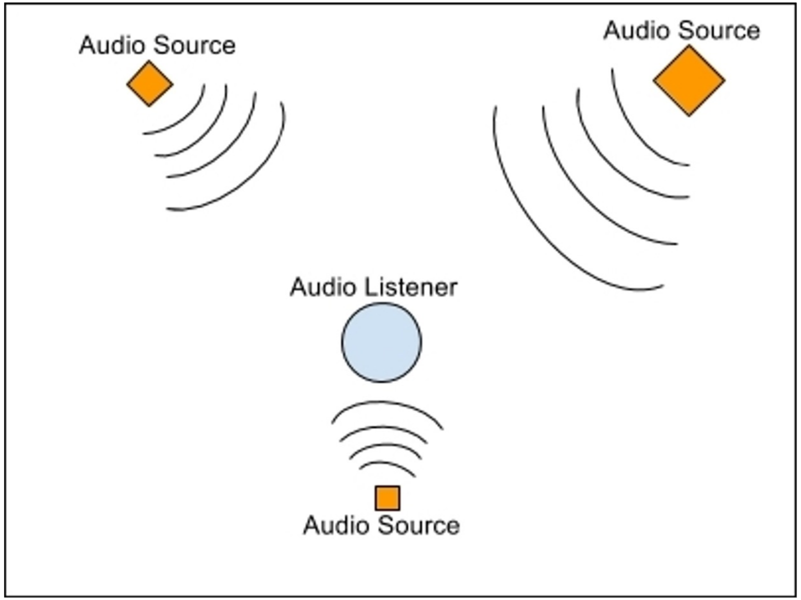
Terminologies & Concepts You Should Know

Some Basic Terms & Definitions

3D Audio Modelling: Soundscape

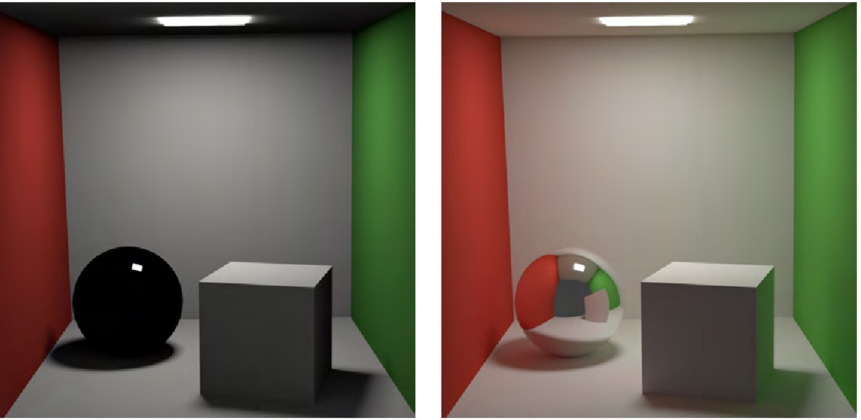
- **Source**: the point from which sound is coming
- **Listener**: the position where the sound is “heard” (in games, usually the Camera)
- **Objects**: things cable of blocking/reflecting sound
    - Can reflect, absorb, or diffuse sound (much like with light)



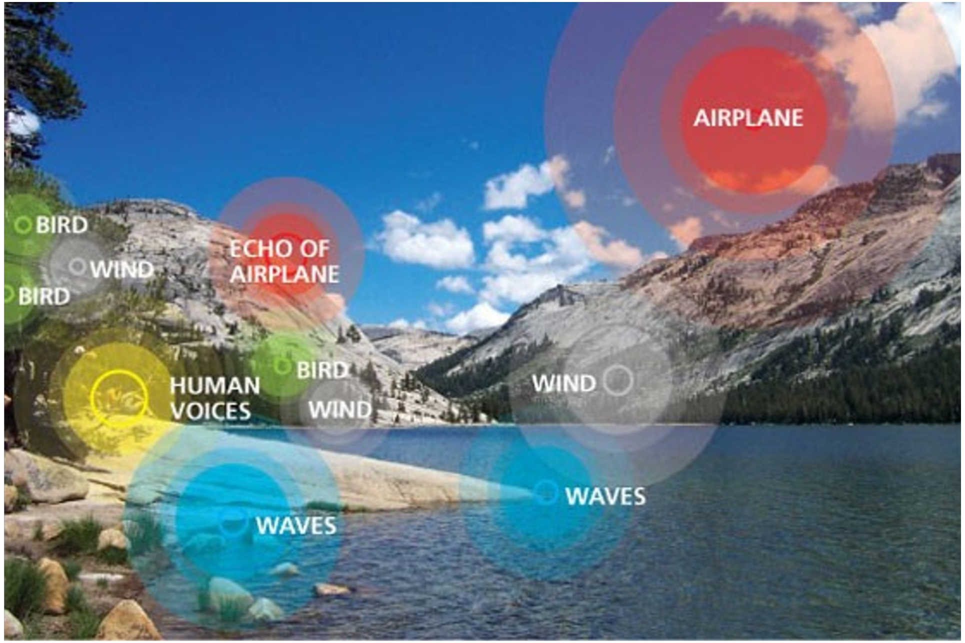


Global Illumination ~ Soundscape

- What VE looks/sounds like from user perspective
- Modelling of acoustic/lighting environment



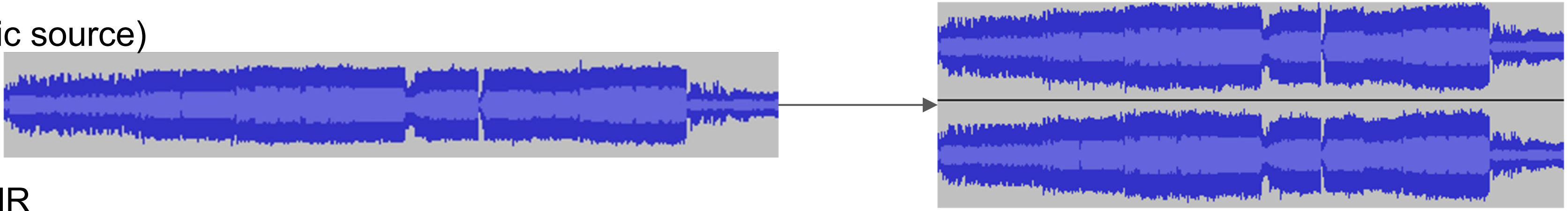
(a) Local Illumination (b) Global Illumination



Spatialized 3D Audio vs. Non-Spatialized

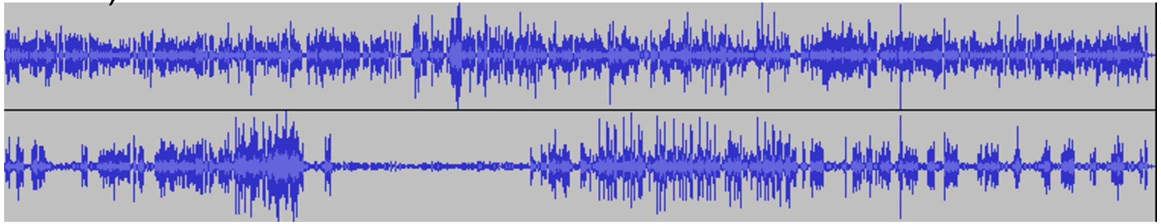
- Mono vs stereo sound:
  - **Mono**: audio sounds same for both ears

(music source)

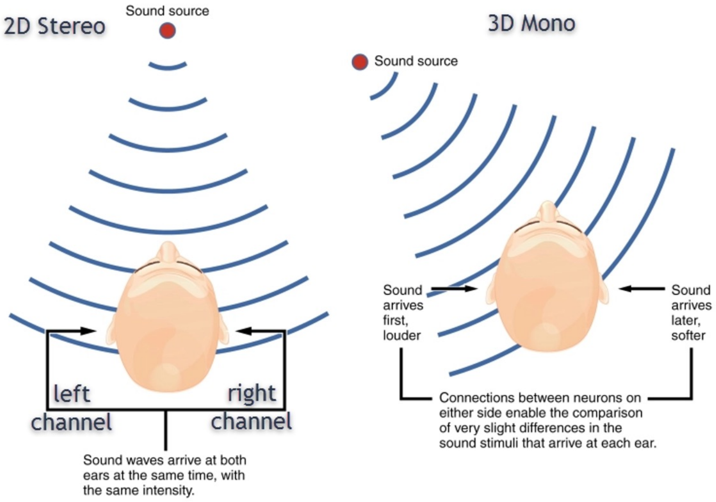


(ASMR source)

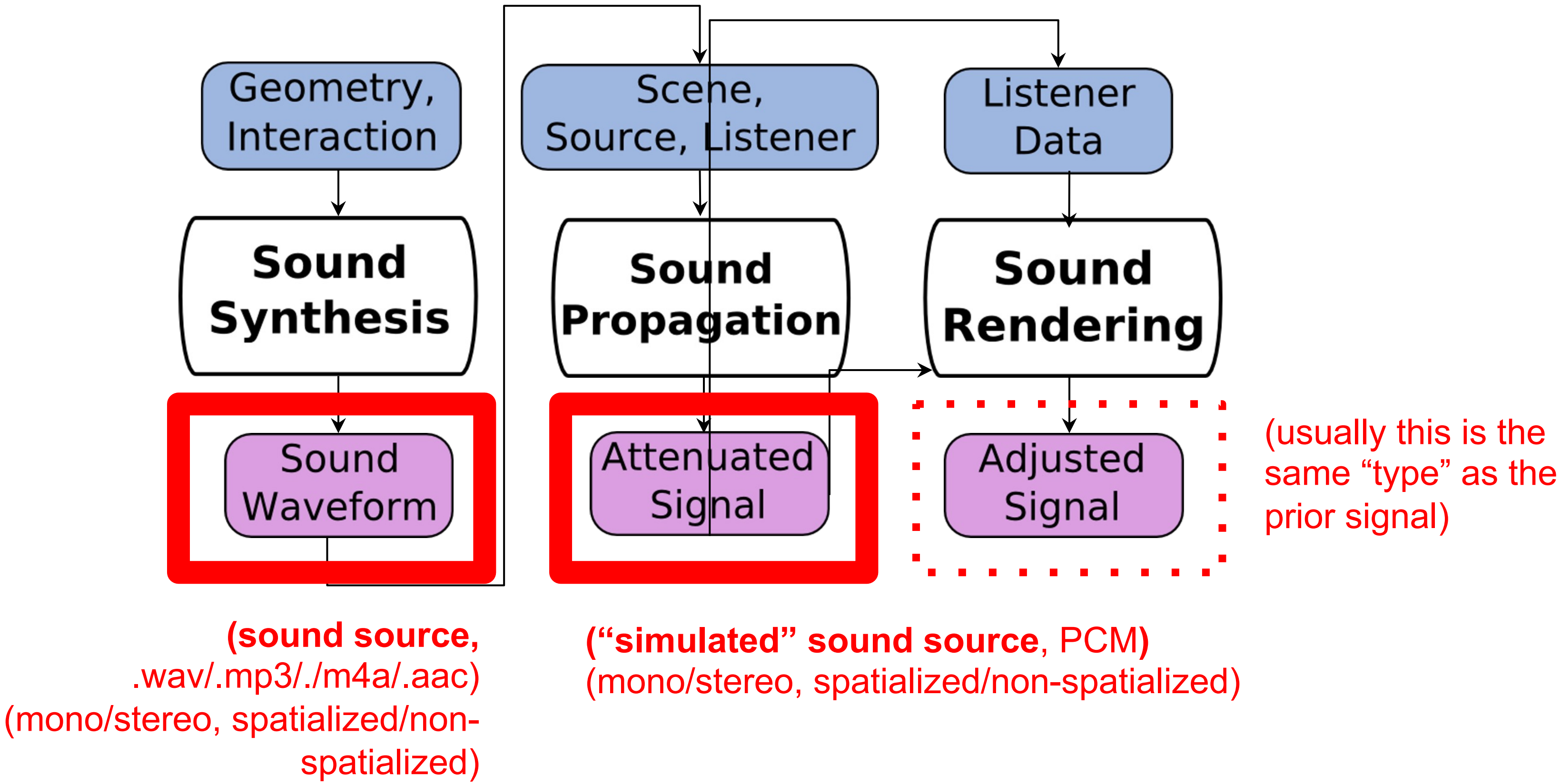
  - **Stereo**: audio can differ between ears (2 channels)



- **Spatialized (3D)**: You can tell where sound is coming from; simulating a virtual acoustic environment
- **Non-spatialized (2D)**: Sound is not simulated in acoustic environment

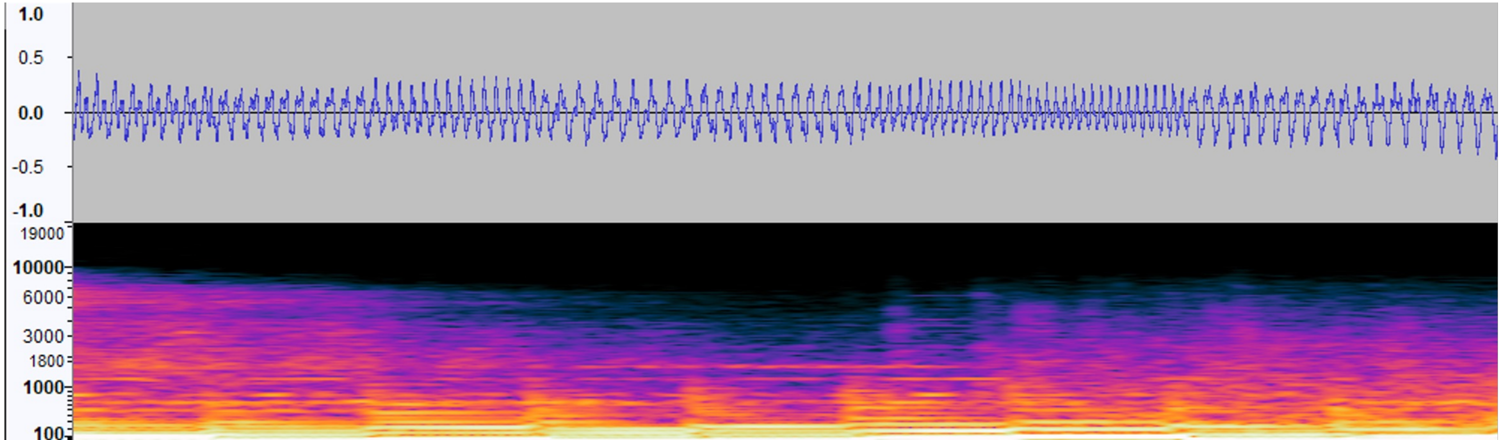


Mono/Stereo, Spatialized/Non-Spatialized Applies to Source Audio AND Runtime Output



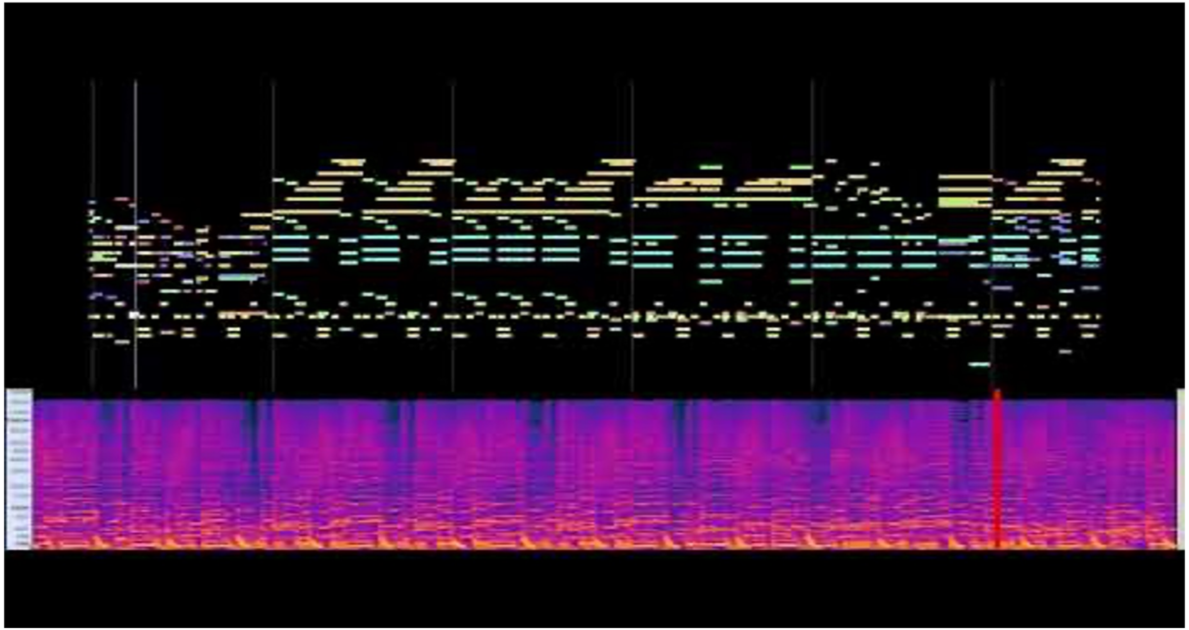
Frequency

- **Waveforms**: ΔdB (signal strength) of air pressure over time
- **Spectrograms**: dB of each frequency/pitch over time
  - Various scaling options affect utility for different applications; e.g. Mel scale
- Often used to filter noise types
  - Low-frequency/**bass**: whale sounds, background noise
  - Mids: human voice
  - Highs/**treble**: hi-hats, bird chirps, snapping fingers, etc.
- (more info in a signal processing class)



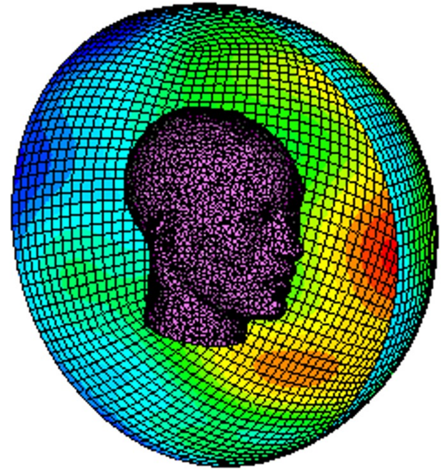
Audio Unmixing

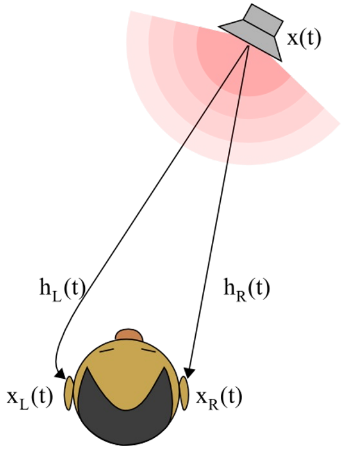
- Audio source gives us a spectrogram… how do we figure out source components?
- Applications in voice recognition, music decomposition, denoising, etc.
- Some ML techniques to figure it out
- New trend: AR tracking used during audio capture to help localize & isolate

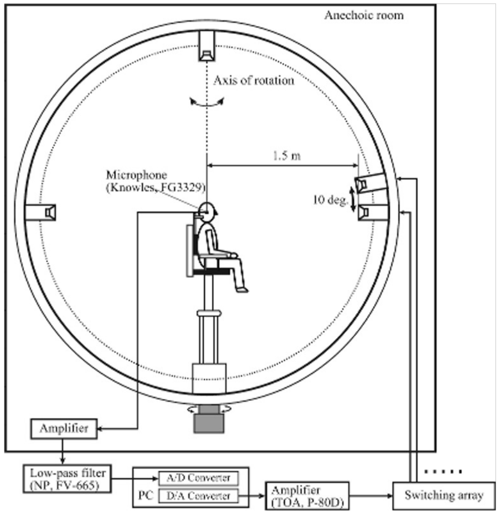


HRTF: Modelling the Ear

- Accurate 3D audio requires us to estimate how sound will bounce through the ear
- *Head-related transfer function* (HRTF) lets us do this
- HRTF estimates sound from particular point travels through ear & head
- Very tedious to generate (use spinning array of speakers and microphones to estimate head shape effect on “hearability”), but there are many libraries of “generic” HRTFs

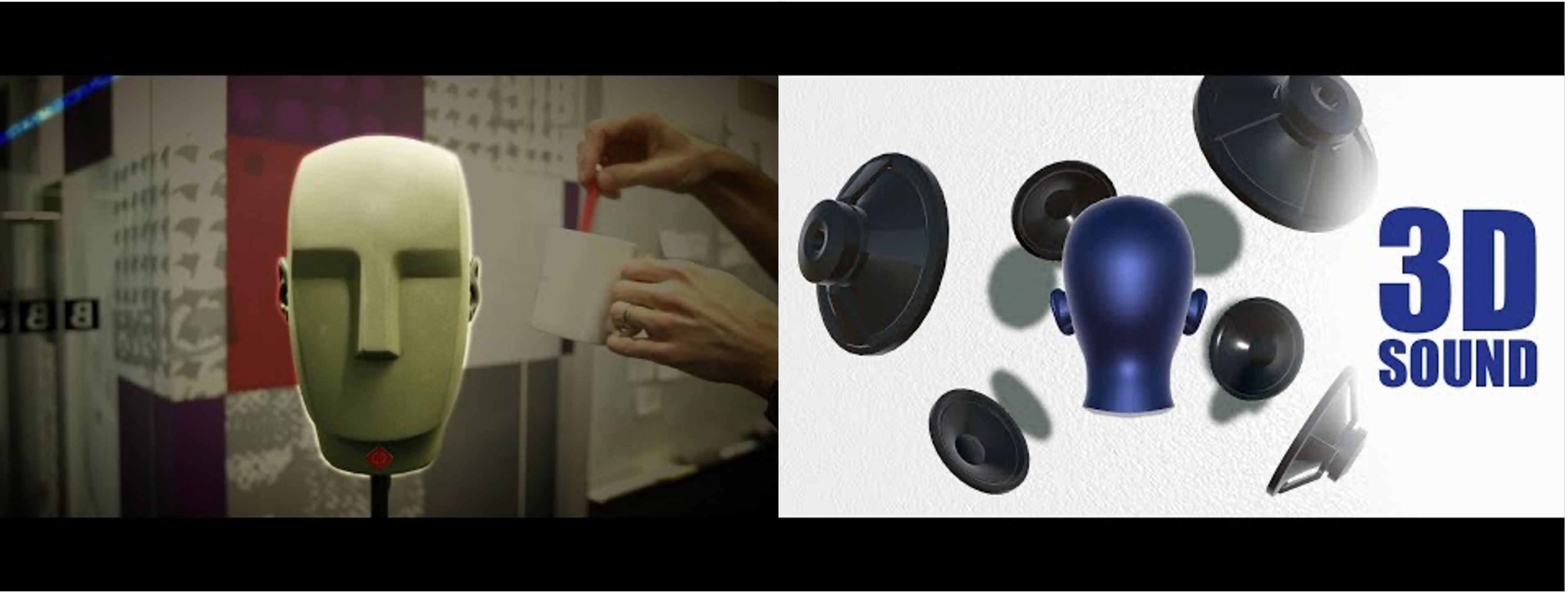






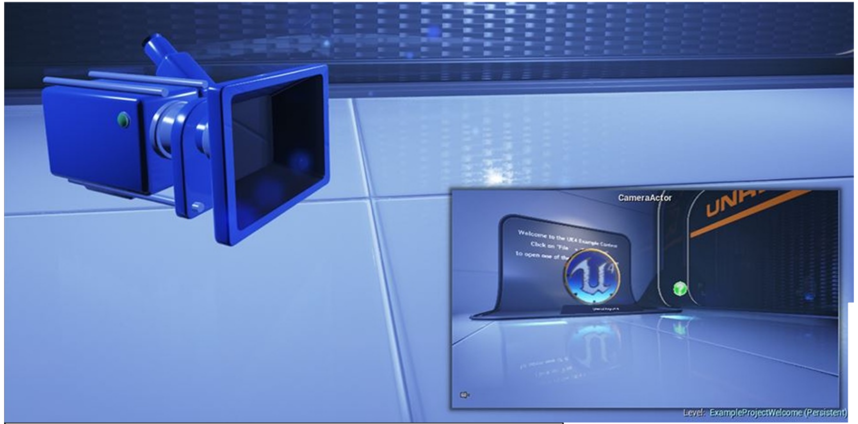
HRTF: Modelling the Ear

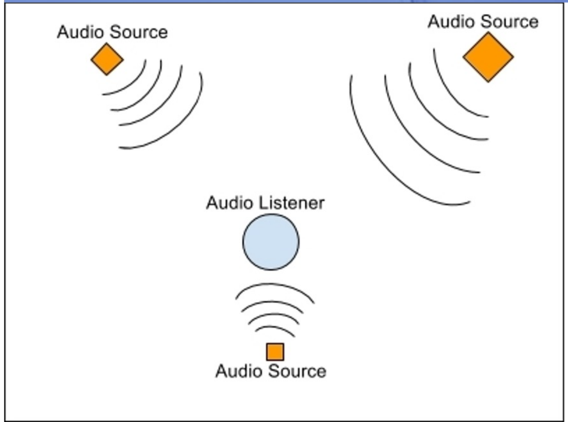
More HRTF

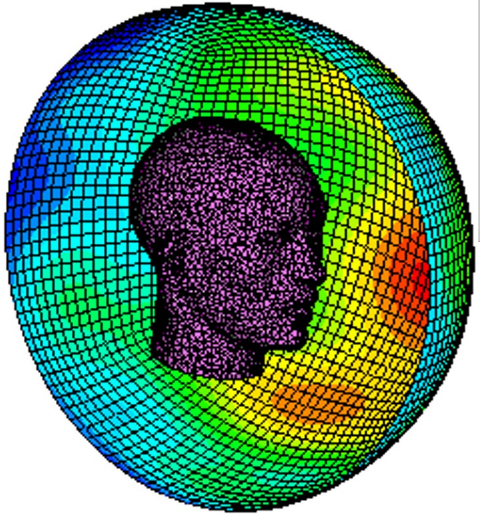


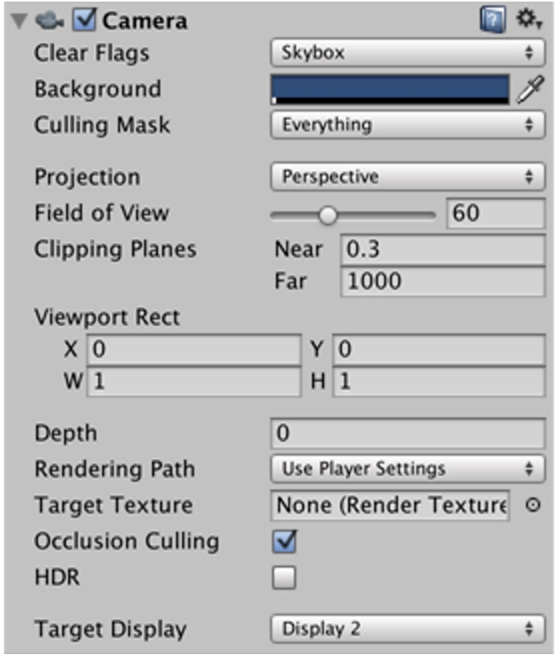
Camera ~ Listener

Both describe how the user hears/sees the rays



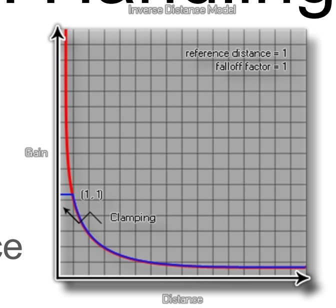


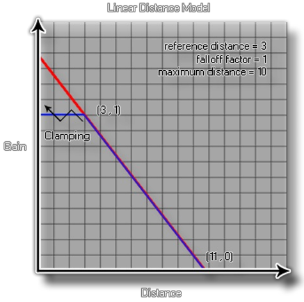


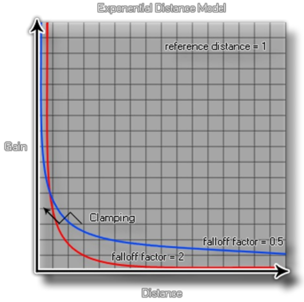


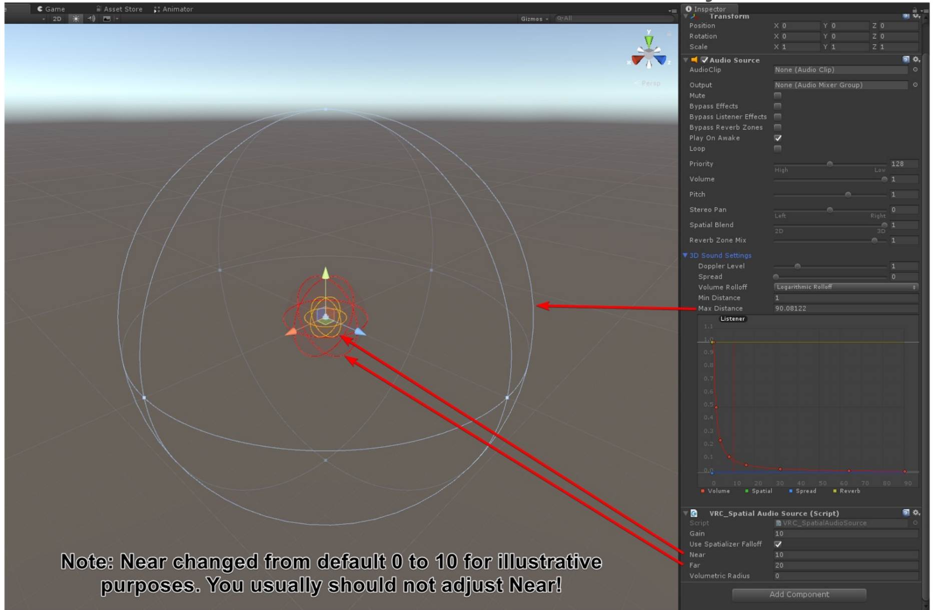
Falloff/Rolloff/Attenuation: Handling Distance

- Volume at various distances
- Logarithmic is common
  - Very loud when right next to sound
  - Levels out to pretty quiet after short distance
  - Decently preserves correct distance; estimate of source distance usually accurate



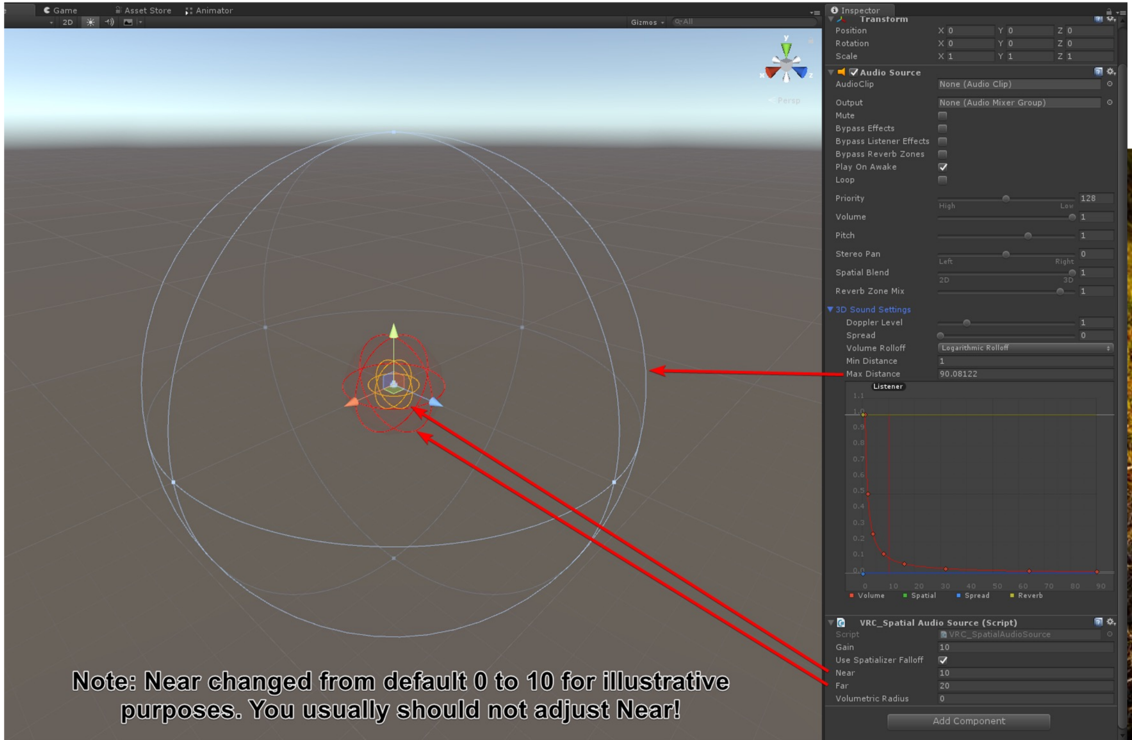






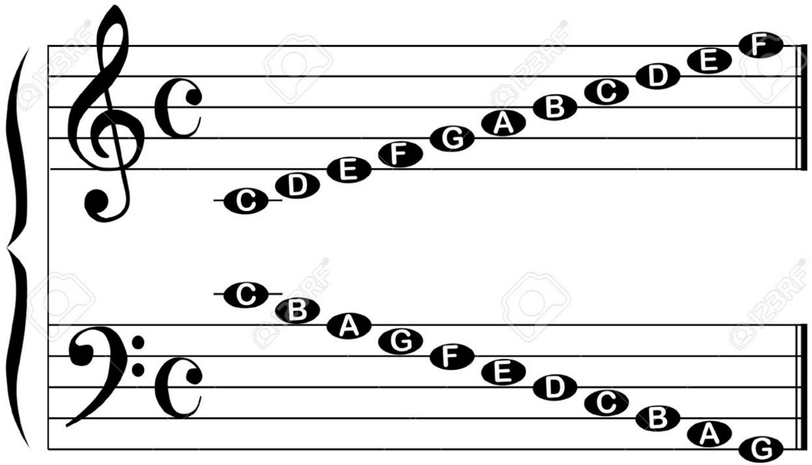
Falloff ~ Depth-of-Field

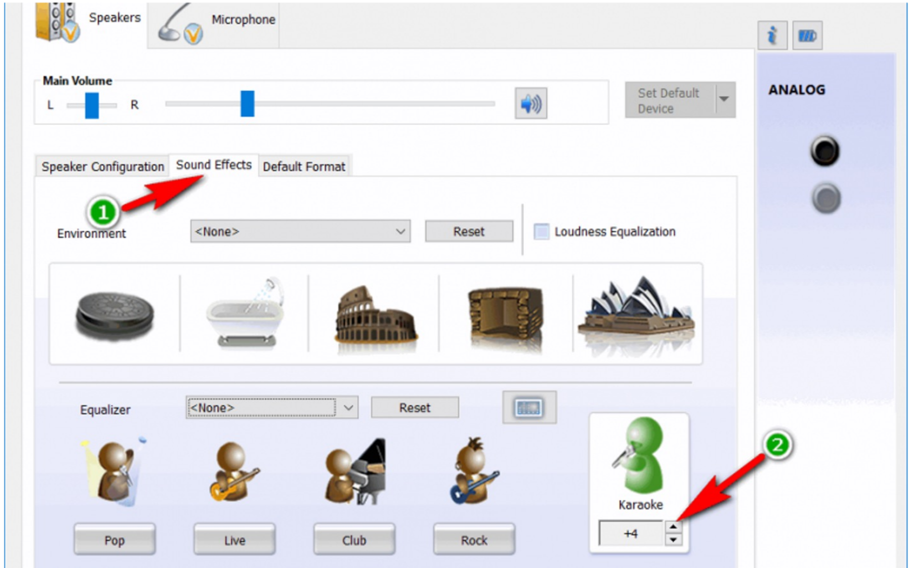
Both describe how visible/audio objects are based on distance (control the “perceived” depth/distance)



Filters

- Controls sound that survives to output
- Usually achieves effect/fakes audio environment
- Often mix of low-pass filter (LPF) & high-pass filter (HPF)
    - LPF: allow low-pitch sounds through but suppress high-pitch (achieves muffling sound) (focus on bass)
    - HPF: opposite (allow high-pitch sounds, suppress low-pitch) (focus on treble)
- Very fast but not always accurate (esp. For dynamic environments)
- Lot of setup & parameters

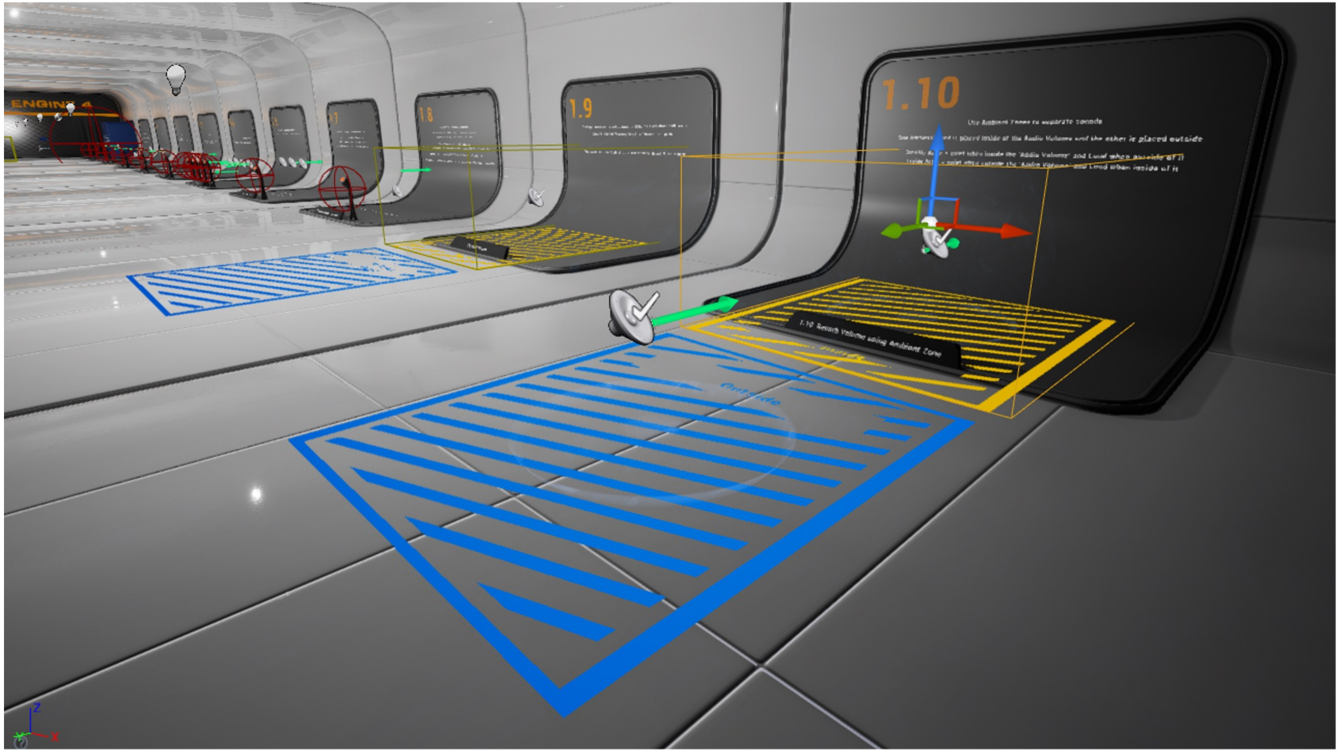




Good video on filters

In 3D...

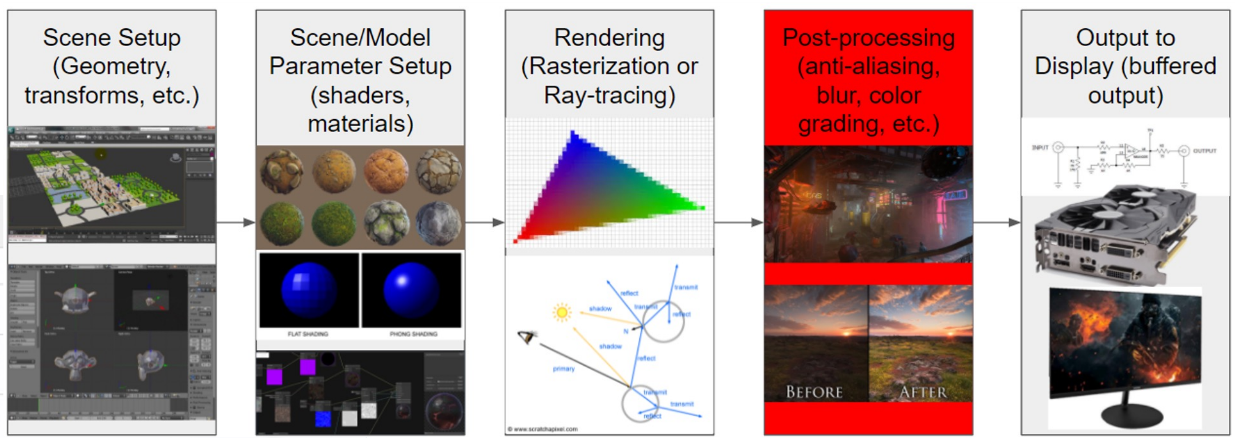
- Filters often applied to “volumes”
- If listener is within that volume, that filter is applied
- Check out UE4 Content Example project for nice examples

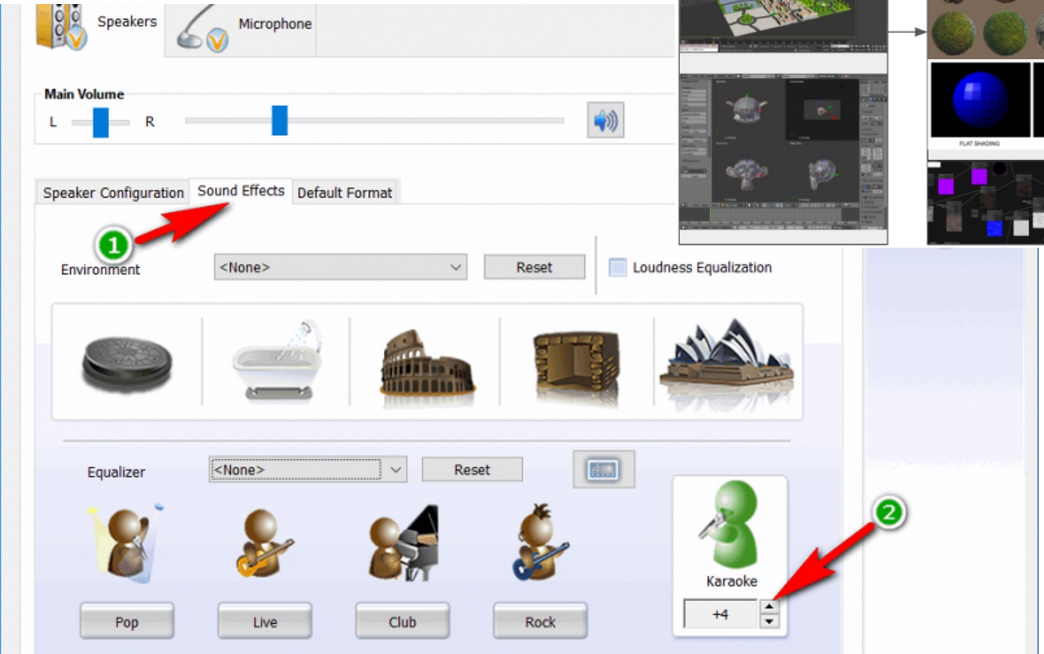


Filters ~ ???

Filters ~ Post-Processing

Fast edits made to source/frame after the bulk of rendering/ “composition” is done



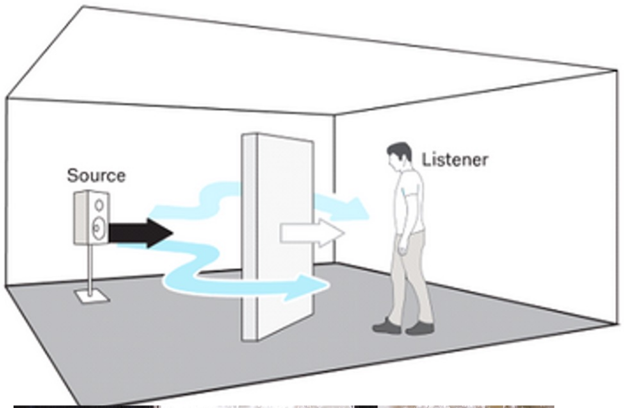


Occlusion

- Sound when something is in front of it (occluding)
- Often achieved with raycast+low-pass filter or propagation (soon to come)

Occlusion

- In light, some materials (e.g. glass) allow light to pass through
- In audio, some materials allow some sound to pass through (e.g. muffled)

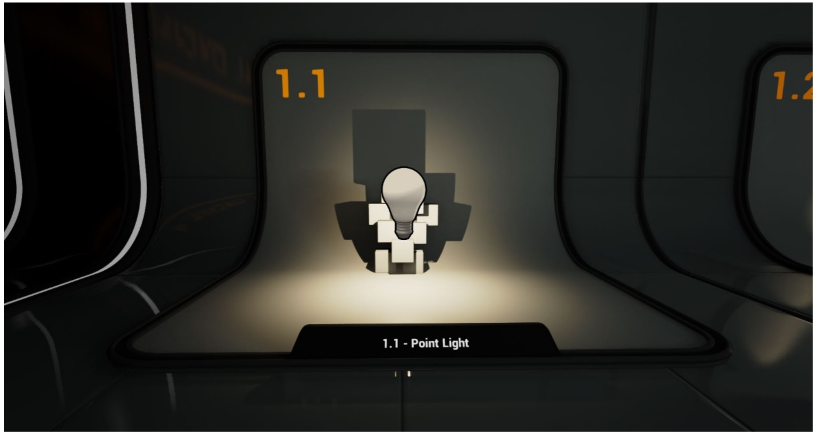


Directionality

- Can have types of audio sources like light sources… point, directional, etc.
  - Point: sound radiates uniformly; sounds same from all directions in a vacuum (at same distance)
  - Directional: much louder when in front of the source (e.g. where sound is coming from, like front of radio)

Audio ~ Light: Types of Sources

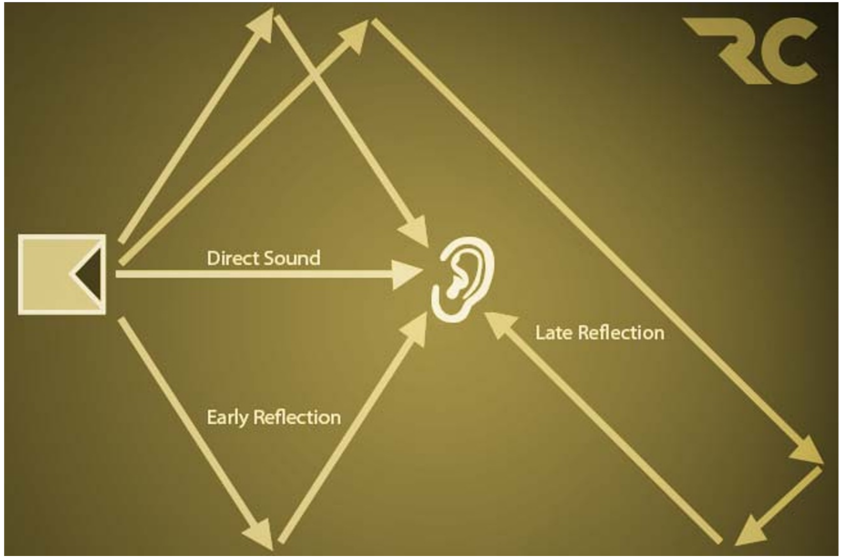
- **Point**: radiates uniformly
- **Directional/Spotlight**: weighted towards direction
- **Ambient**: uniform strength applied to all surfaces (e.g. background noise/music)
    - In light, nothing is usually pitch black
    - In audio, nothing is usually perfectly silent



Reverberation

- Filtered or propagated
  - **Direct audio**: Sound that's in line of sight
  - **Early reflections**: Sounds that bounce and are heard soon after
  - **Late reflections**: Sounds that bounce and sound very delayed (e.g. echos, long-range, etc.)
- Light has similar concepts
  - **Direct light**: looking at the sun
  - **Early reflections**: objects that you look directly at
  - **Late reflections**: mirrors, stars, reflections, etc.
- Reflection "**order**" is how many bounces until reaches listener

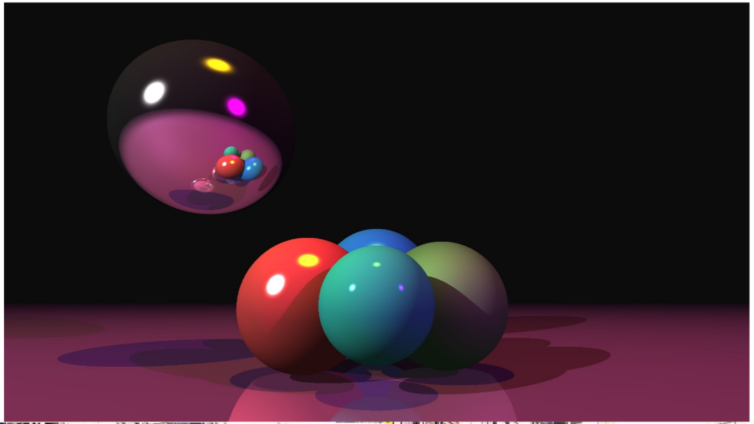
  (light travels much faster and doesn't attenuate much so the ray order doesn't matter as much)

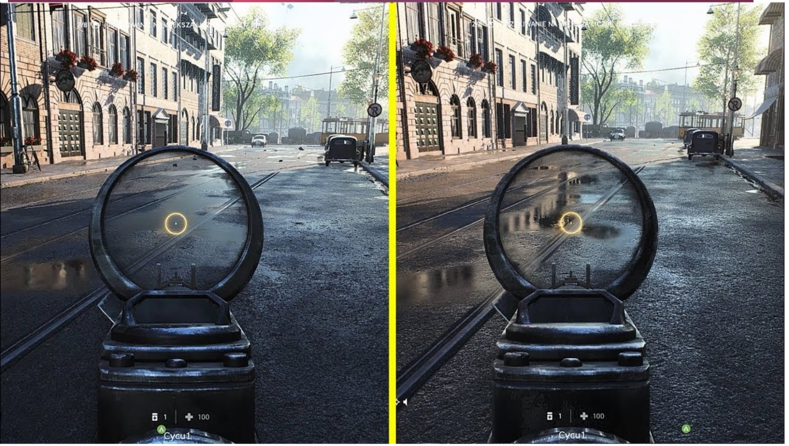


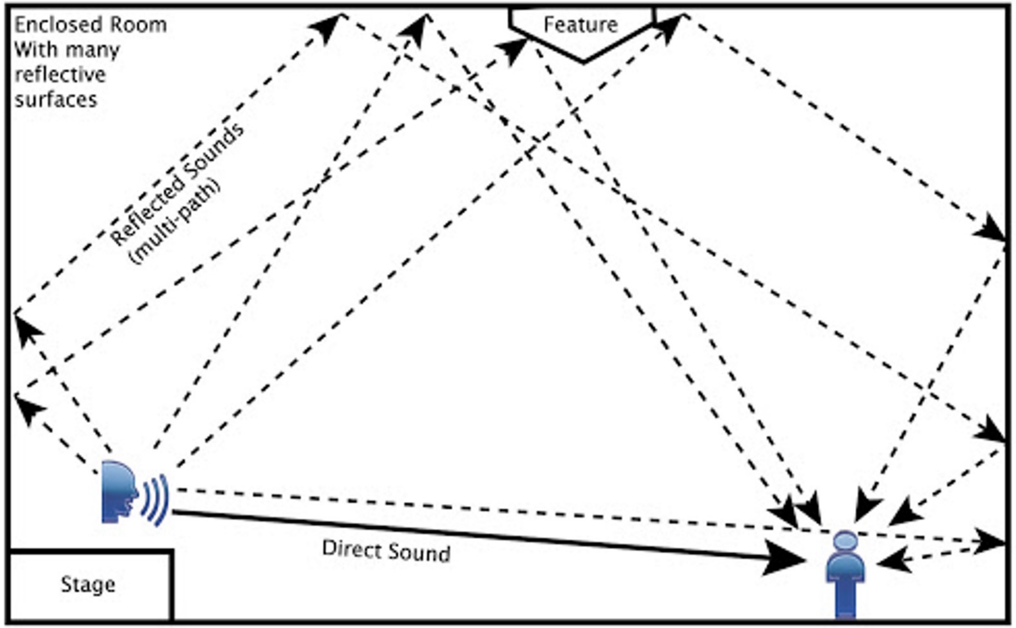
Sound propagation ~ ???

Ray/Path-tracing ~ Sound propagation

- Trace how rays bounce around for more accurate results for things that are reflective

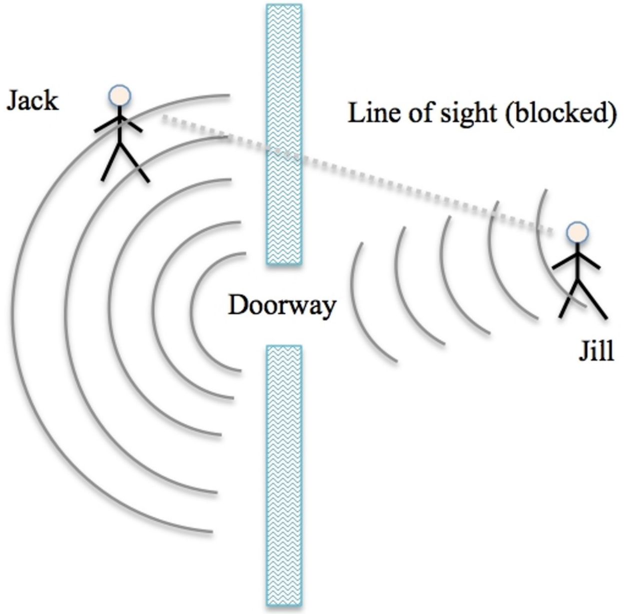






Diffraction

- How a sound is heard when not in line of sight
    - (similar effect to occlusion….except diffraction worries about how the sound wave "bends")
- Often thought of as muffling….but is a bit more geometric than LPF
    - E.g. sound is usually still pretty clear near a boundary and transitions smoothly



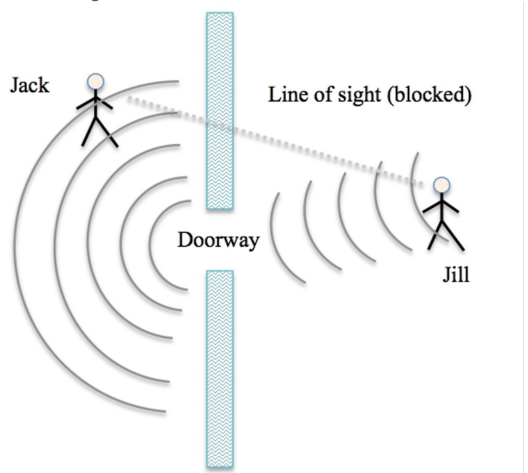
Diffraction Example

Diffraction Example

Diffraction ~ ???

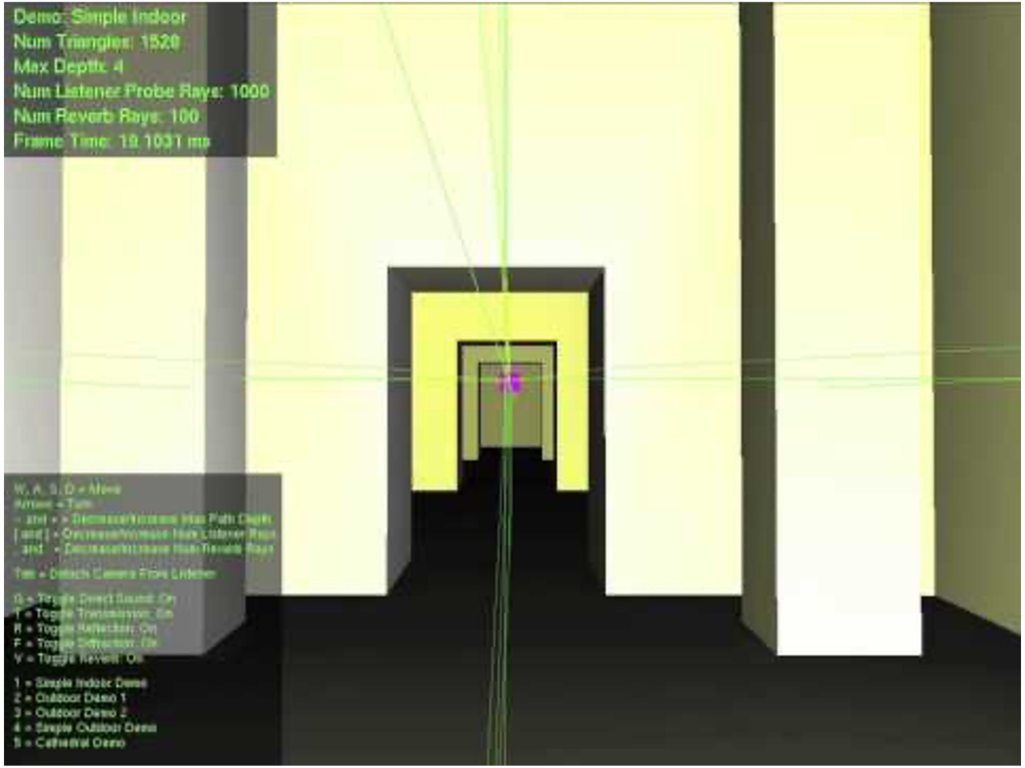
Diffraction ~ God Rays

- Distort light/sound around an edge…. Distorted rays can bounce around



Sound Propagation (Ray-Traced Audio)

Great video from UNC (by people now at Oculus)

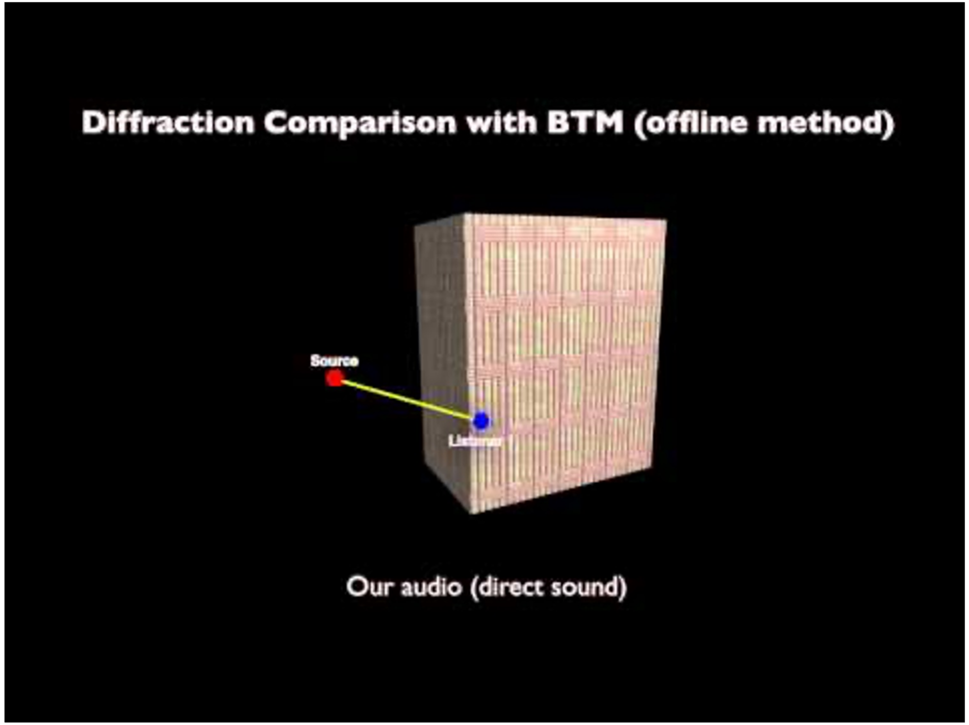


Sound Propagation (Ray-Traced Audio)

Great video from GAMMA (by people now at Oculus, Microsoft, Valve)

Even more


Numerical
100
6


Diffraction Comparison with BTM (offline method)
Source
Our audio (direct sound)

Reverb Zone Parameters

- In real life, low-freq sounds echo for much longer (don’t die as quickly)
- **Density**: how many rays are traced (fullness of the reverb)
- **Diffusion**: how far the bouncing rays get from each other
  - So more diffuse for environmental sounds, less for voices
- **Reflections Delay**: how long it takes ray to hit objects/time between echoes
  - Also affects how long it takes reflection to reach listener
- **Gain**: strength of source signal (volume before first bounce)
  - **Gain HF**: signal strength for high-freq sounds (e.g. should be higher for voices to travel)
- **Decay Time**: how quickly the sound attenuates/weakens
  - **Decay HF**: Same for high-freq sounds (high-freq sounds usually die faster)
    - Described as percent of decay time (e.g. <1 means HF lasts shorter)
- **Reflections Gain**: How much volume survives after hitting obstacle
- **Late Gain**: Above but for late reflections (e.g. after multiple bounces)
- **Late Delay**: Same as above but for late reflections
- **Air Absorption Gain HF**: How much air absorbs HF sounds
- **Room Rolloff Factor**: Size of reverb zone * attenuation

(so normalizes rays based on room size)

A pretty good resource (not exactly game engine params): https://www.emusician.com/gear/cheat-sheet-reverb-parameters

Also recommend playing with the UE4 Content Examples Audio map

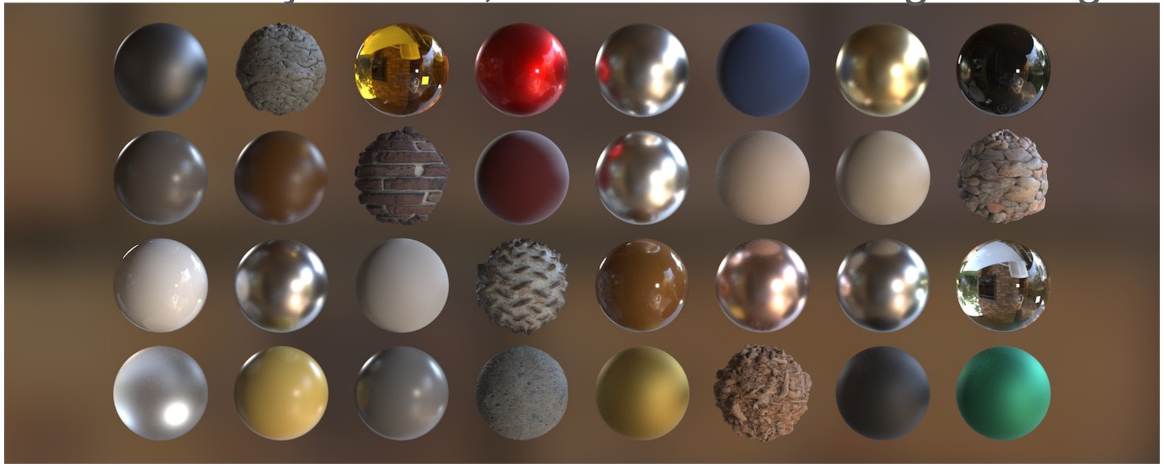
Material Modelling

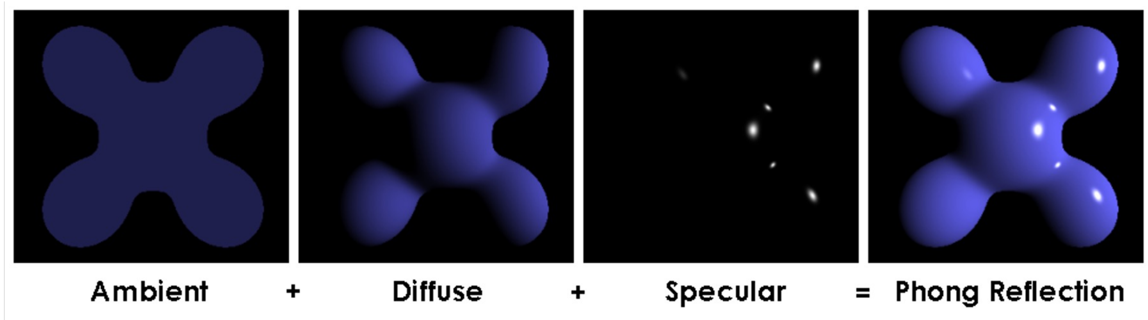
- Modern sound APIs model the materials of the room & objects
  - E.g. metal has a lot of reverb, carpet absorbs most reflections

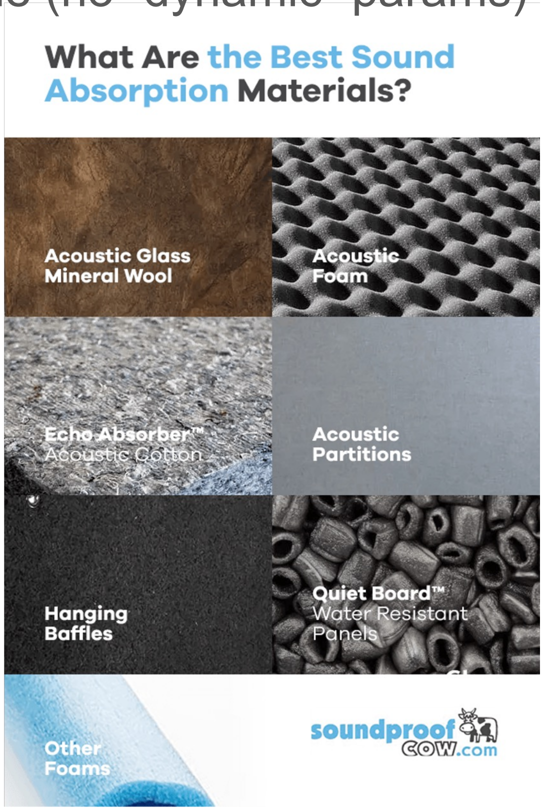
Acoustic Materials ~ ???

Phong materials ~ Acoustic materials

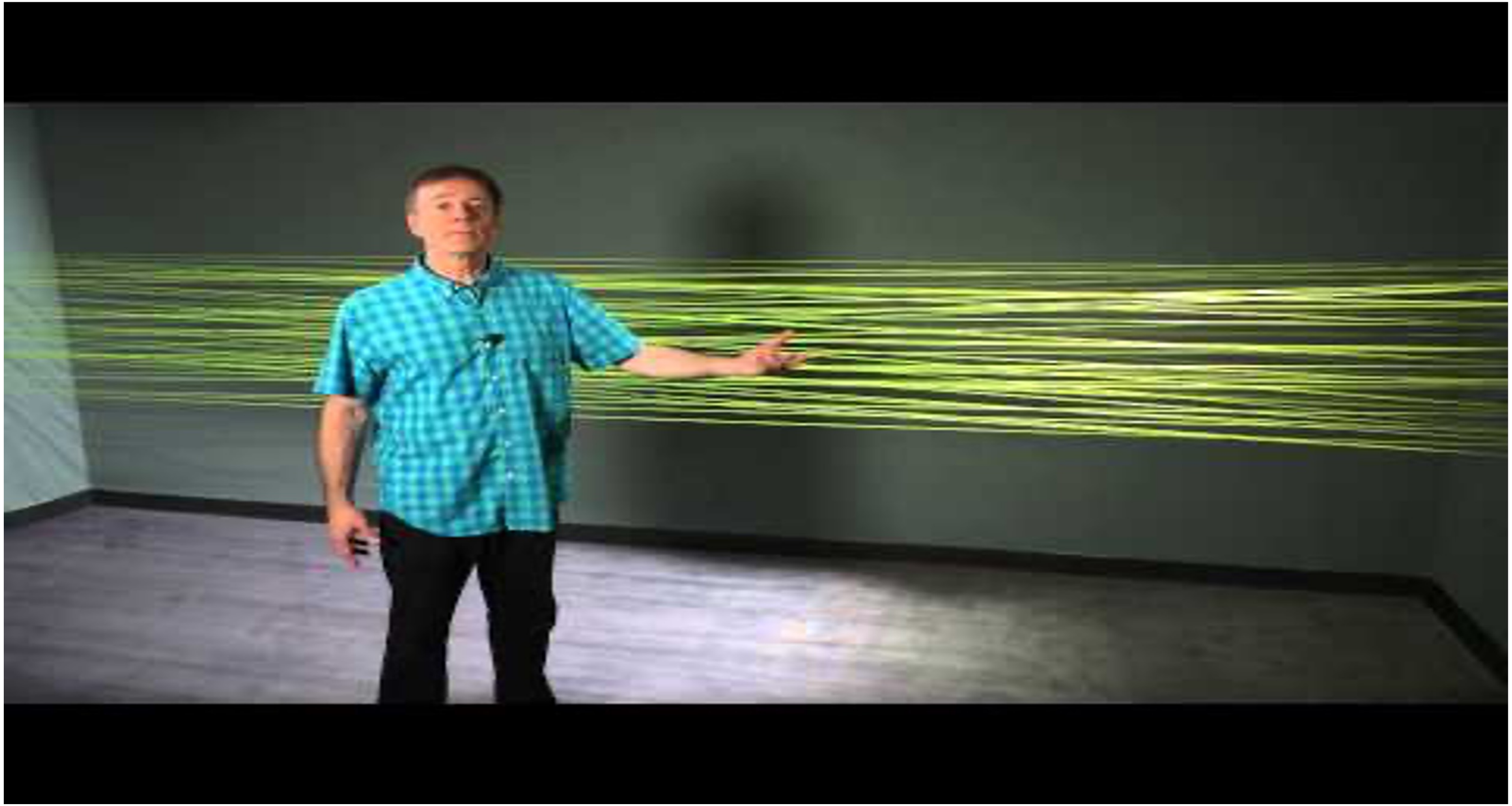
- Both contain scalar parameters describing how that surface affects rays
- Mostly “baked”; not much will change during runtime (no “dynamic” params)





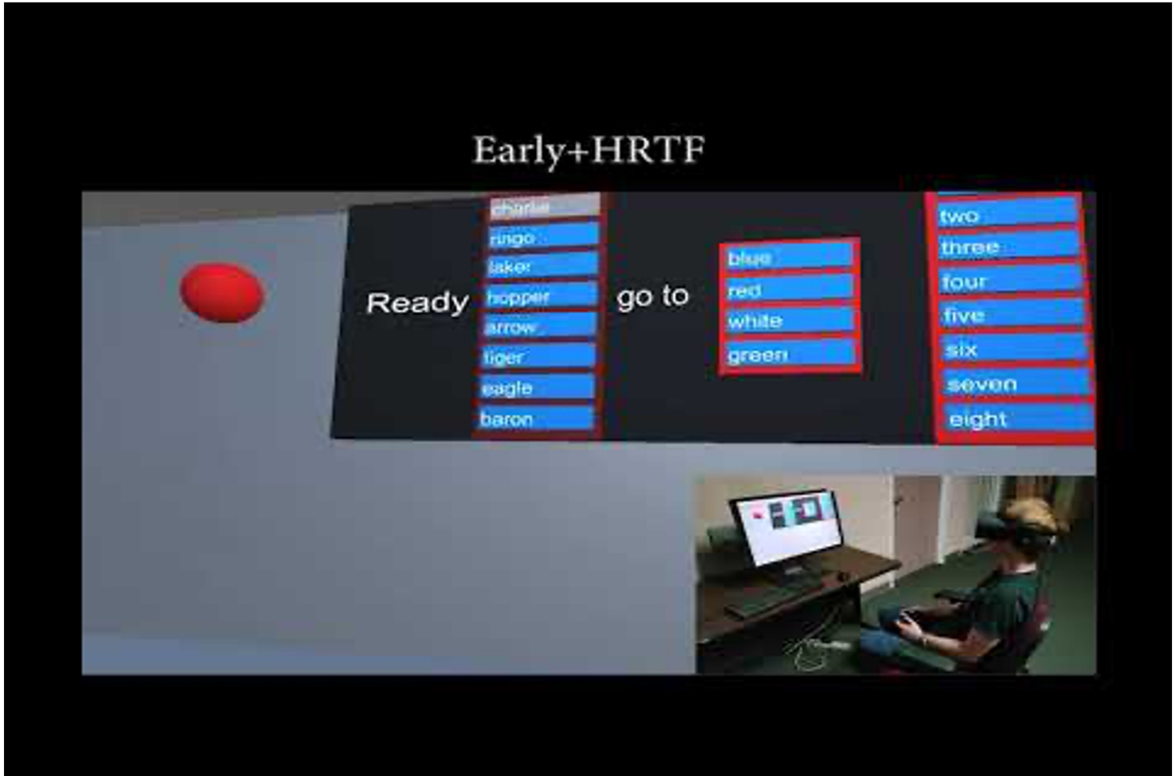


Video of how geometry affects sound



Other Perceptual Illusions Apply to Sound

- E.g. distance compression (sound usually sounds way too close or far)
- E.g. Cocktail party effect (hearing specific audio cues when lot of sources)

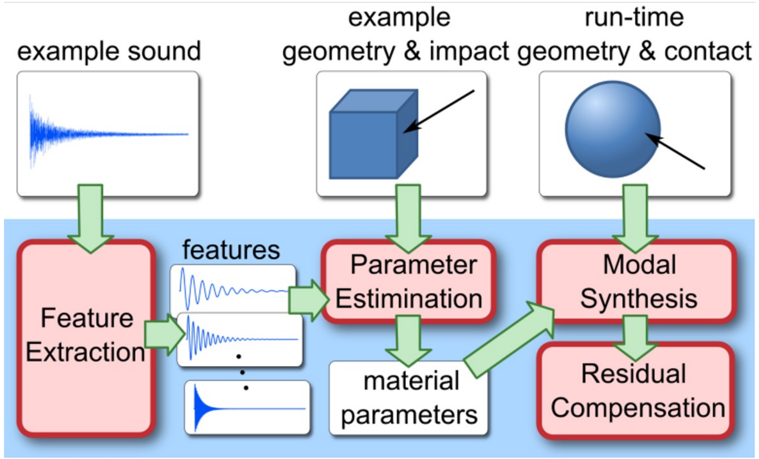


APIs for game engines

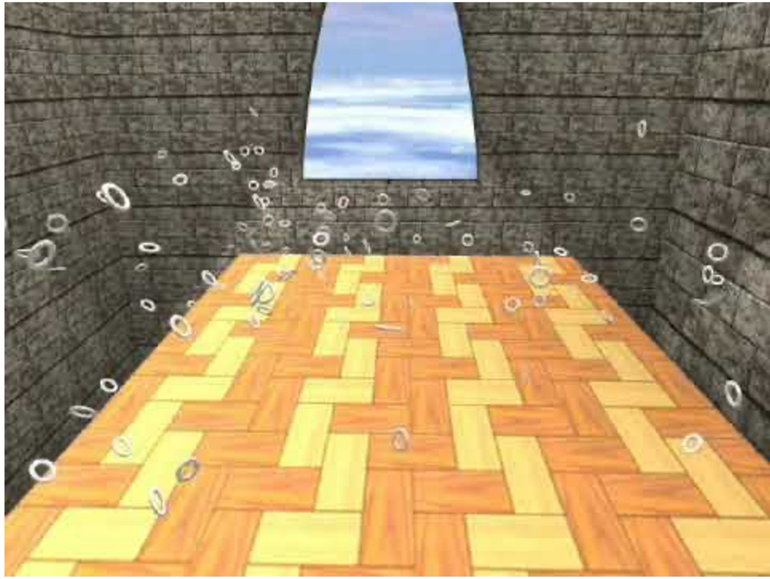
- **Google Resonance**: doesn't seem to have geometric propagation but lower requirements/works better on mobile, open source
- **SteamAudio/Phonon**: IMO easiest to use, best documentation, has good geometric propagation, open source
- **Oculus Audio**: similar to SteamAudio, also has geometric propagation, has more and better preset materials/filters IMO
- **GSound/MSound**: Predecessor to many propagation libraries by GAMMA group
- **UE4 & Unity built-in audio:** Have built-in:
  - Spatialization
  - Filters
  - Occlusion
  - Reverb zones
  - Procedural sound/propagation

Sound Synthesis (Physically-Based Audio)

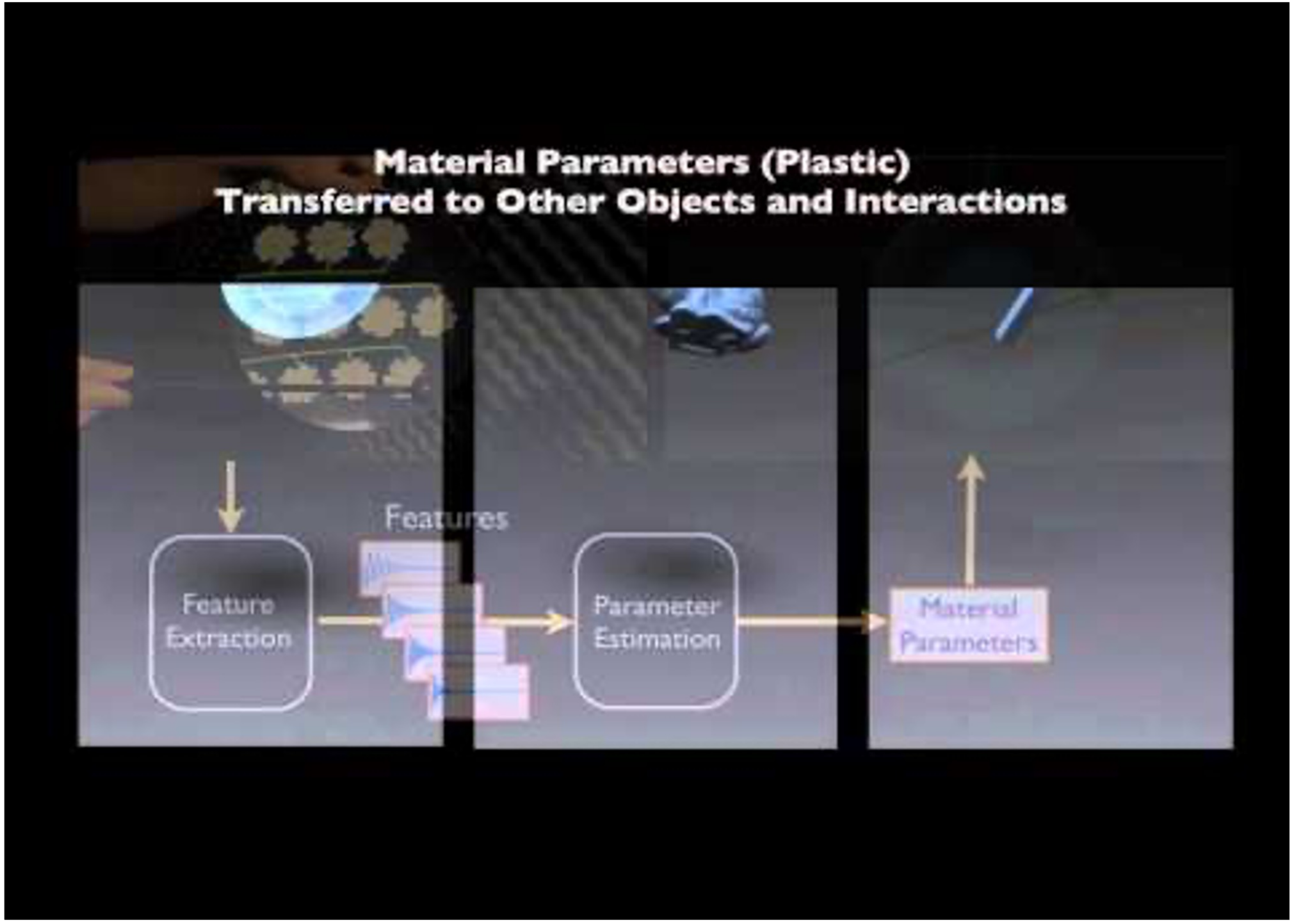
- **General idea**: use geometry & material params of mesh to figure out correct sound on impact
- Even less accessible today than propagation; still very hard to do realtime
- Common method: **modal sound synthesis**: figure out “modes”/features of the audio (e.g. glass, metal, etc.), apply those modes to other geometry, & simulate
  - So once you know modes, you can use a physically-based material on any mesh



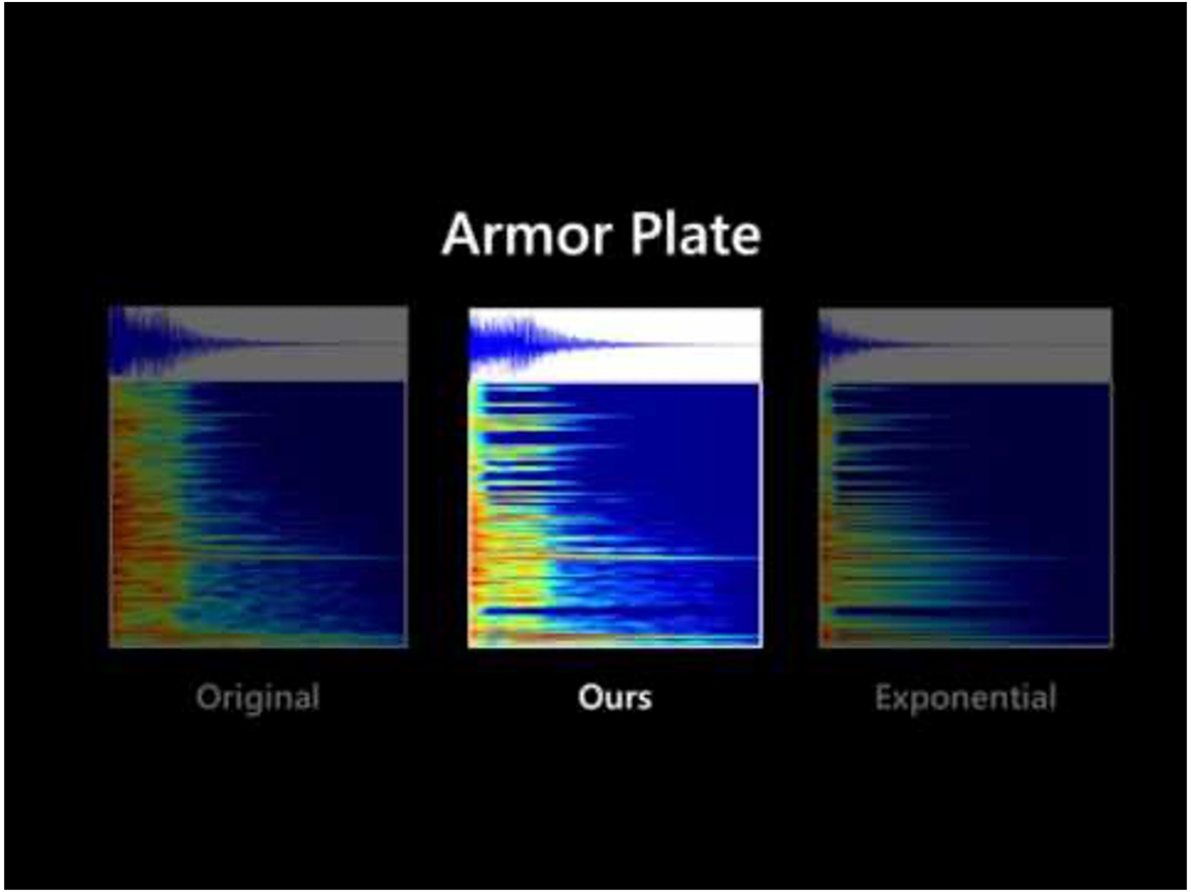
From “Example-Guided Physically Based Modal Sound Synthesis”



Modal sound synthesis example



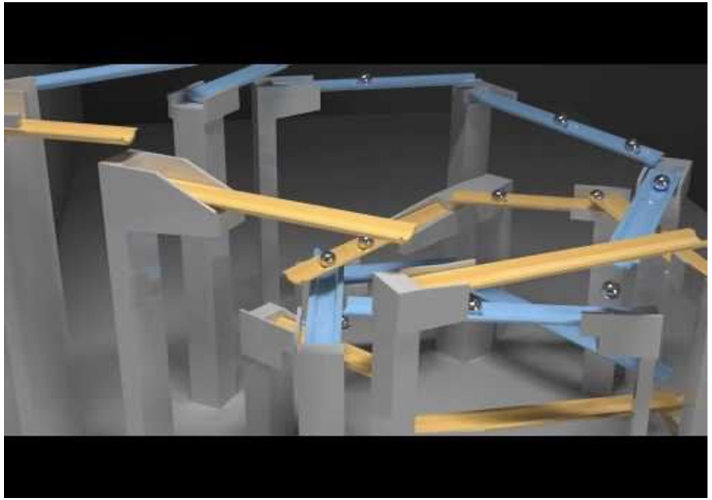
Game-focused modal sound synthesis

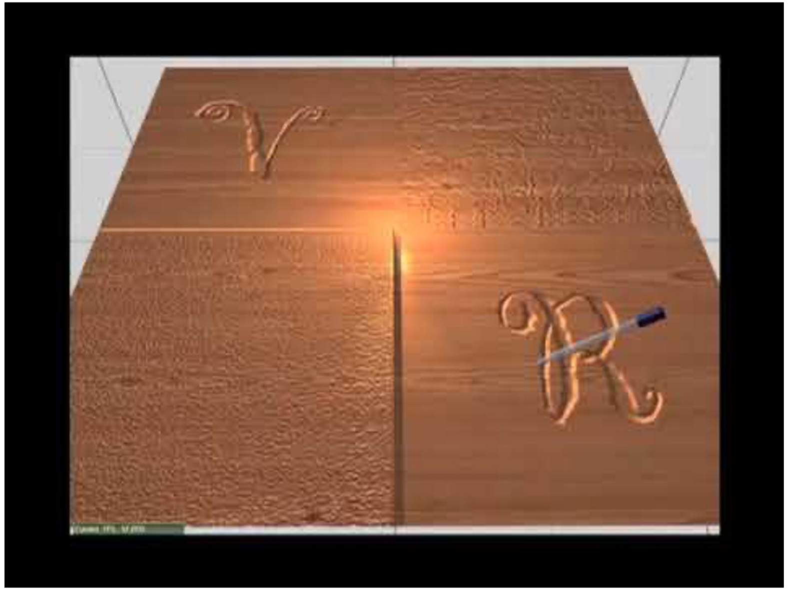


More Advanced Sound Synthesis

Challenges of sound synthesis

- Requires huge precompute step
- Has trouble with continuous contact sounds & damping (multi-object interaction)
- Usually performs poorly realtime depending on how much is precomputed
- Doesn't always respond well to questionable game engine physics
- **But we're getting there!**



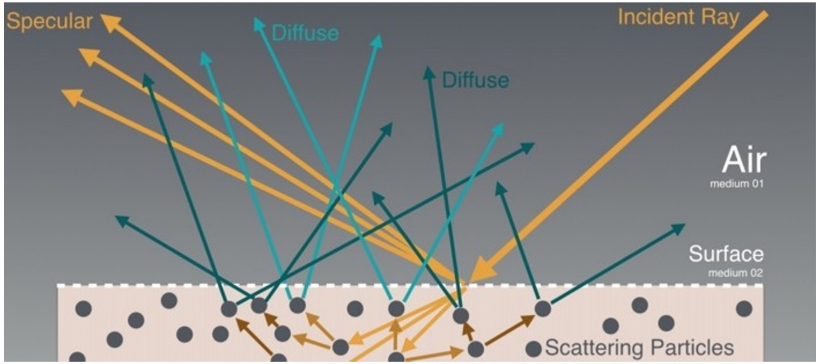


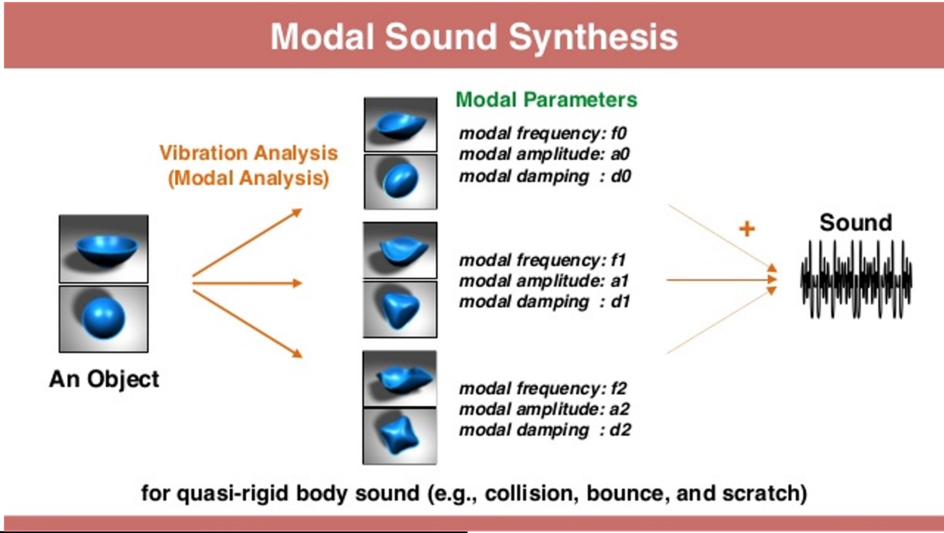
GAMMA Research on VR sound synthesis

Sound Synthesis ~ ???

Physically-Based Rendering ~ Sound Synthesis

- Both generate results realtime starting from material definition + maps/features instead of precomputing/baking them





Questions?