Sound synthesis in Crackdown 2 and wave acoustics for games

Nikunj Raghuvanshi, Brandon Lloyd, Guy Whitmore

Microsoft Research **Popcap games**

Immersion

- Graphics has made leaps and bounds
- Audio far behind: great potential

Image taken from: incrysis.com

Sound in games today

- Recorded samples
- Manually tuned roll-off/occlusion/reverb filters

- Realism requires tedious labor
- Reduces time for creativity

Physically-based sound

- Physics — Perception

automation — “Physically-based” — control

- Use auditory perception → compact, parametric models
  - realistic-sounding starting point
  - artistic control
  - quality-efficiency tradeoff
- More time for creative work

Overview

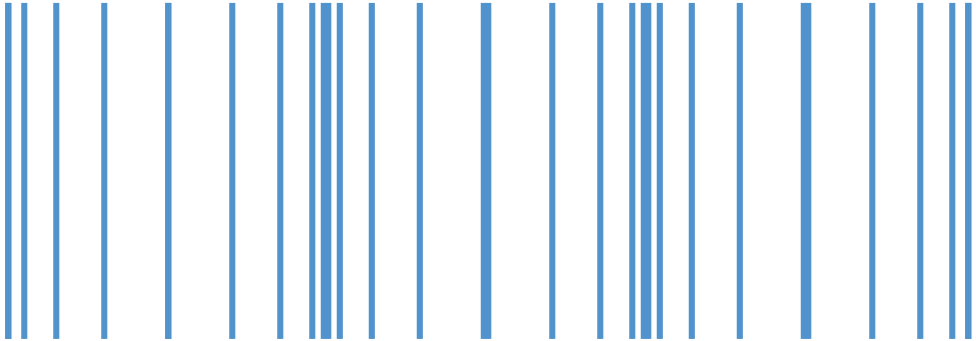
- Physics of sound
- Sound synthesis in Crackdown 2
  - Collaborators: Brandon Lloyd & Naga K. Govindaraju at MSR, and Guy Whitmore & Kristofor Mellroth at MGS
- Wave acoustics
  - Collaborators: John Snyder & Naga K. Govindaraju at MSR and Ravish Mehra and Ming C. Lin at UNC Chapel Hill
- Conclusion

Sound Synthesis

- Collisions lead to surface vibrations
- Vibrations create pressure waves in air
- Pressure waves perceived as sound

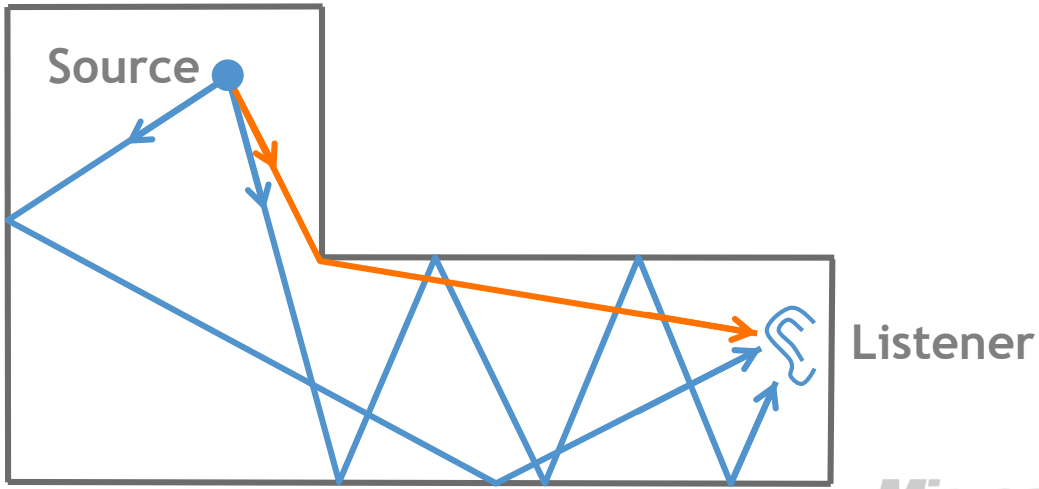
Vibration

Pressure (sound) waves



Sound Propagation (Acoustics)

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



Overview

- Physics of sound
- Sound synthesis in Crackdown 2
- Wave acoustics
- Conclusion

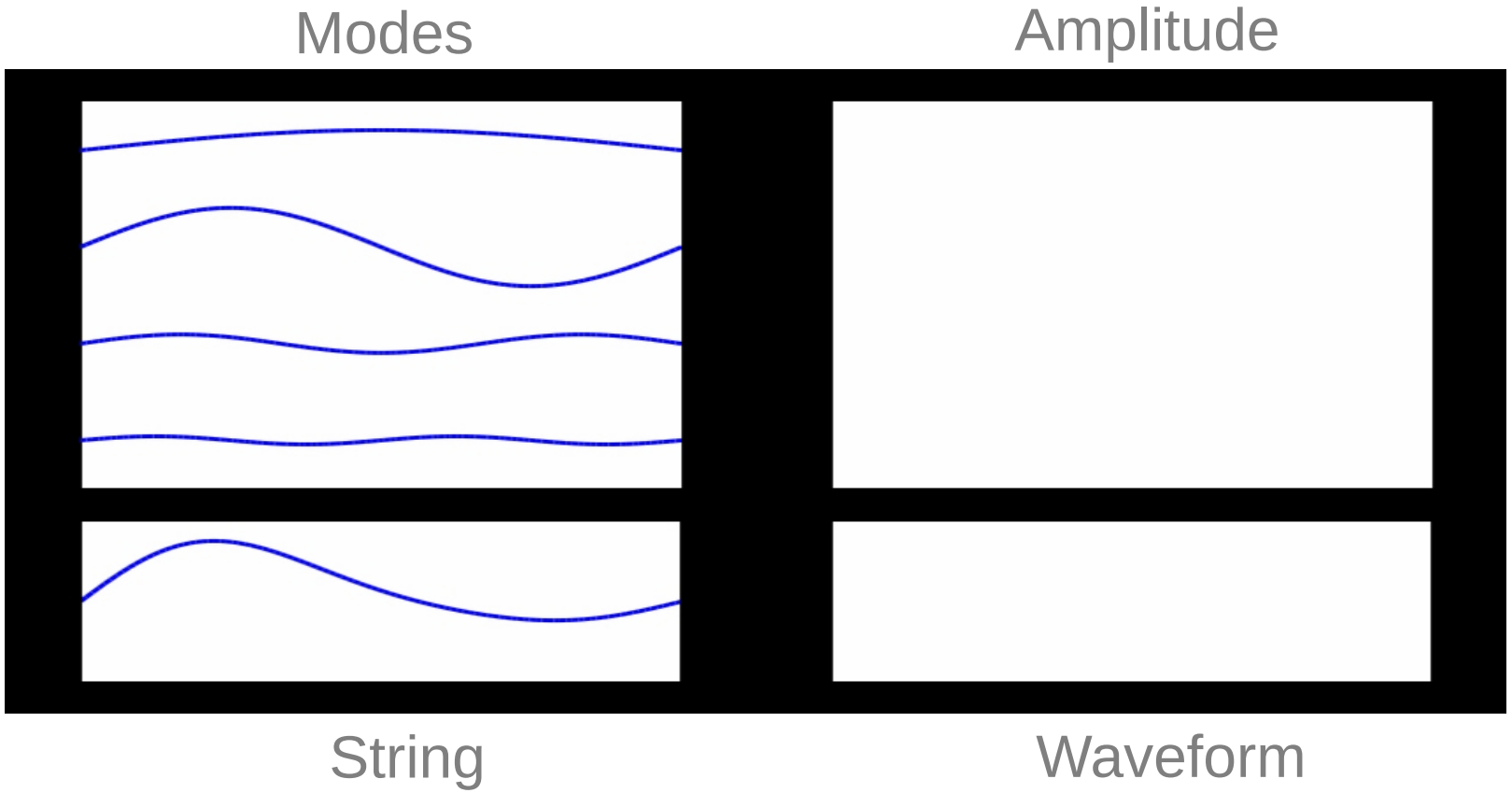
Physically-based impact sounds

- Most games: Variation achieved through multiple clips (3-5) for each sound
  - Inflexible, memory-intensive
- Our approach --
  - Extract compact physical model from a single clip
  - Procedurally generate sound from model (costs compute)
  - **Infinite variation and artistic control**

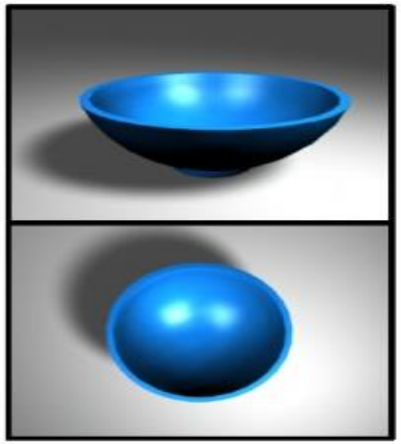
Constraints

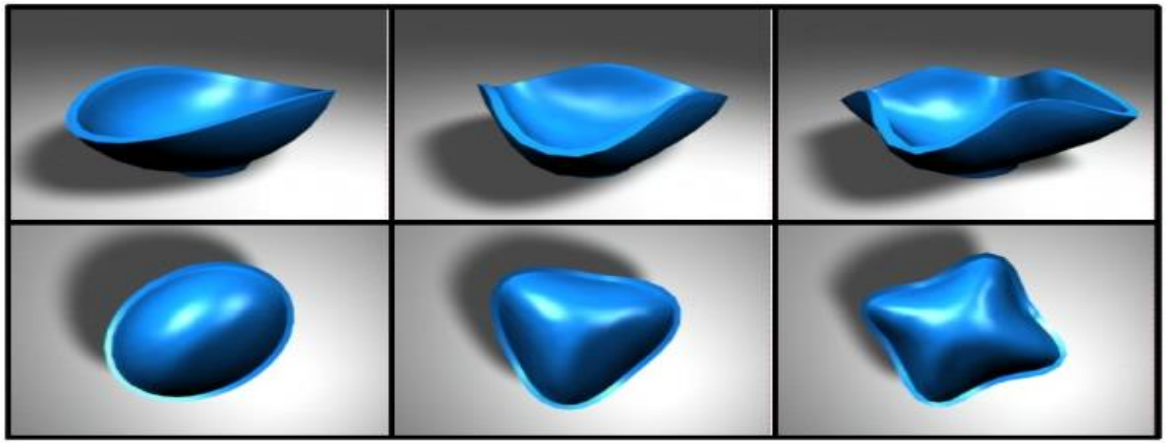
- Limited memory
  - 25 MB for all audio
  - 2 MB for physics sounds
- Bounded CPU usage (10%)
- Easy to use
  - simple pre-processor UI: control efficiency vs quality
  - plugins for AudioKinetic's Wwise

Physical Model: Modal Synthesis



Modal Synthesis





O'Brien et al. 2002

Forward Modal Synthesis

- geometric shape + material → Sound
- Previous work in Graphics:
  - [van den Doel et al. 2001]
  - [O'Brien et al. 2002]
  - [Raghuvanshi and Lin 2007]
  - [Bonneel et al. 2008]
- All assume ideal exponential decay

Forward Modal Synthesis: Pros and Cons

Pros:

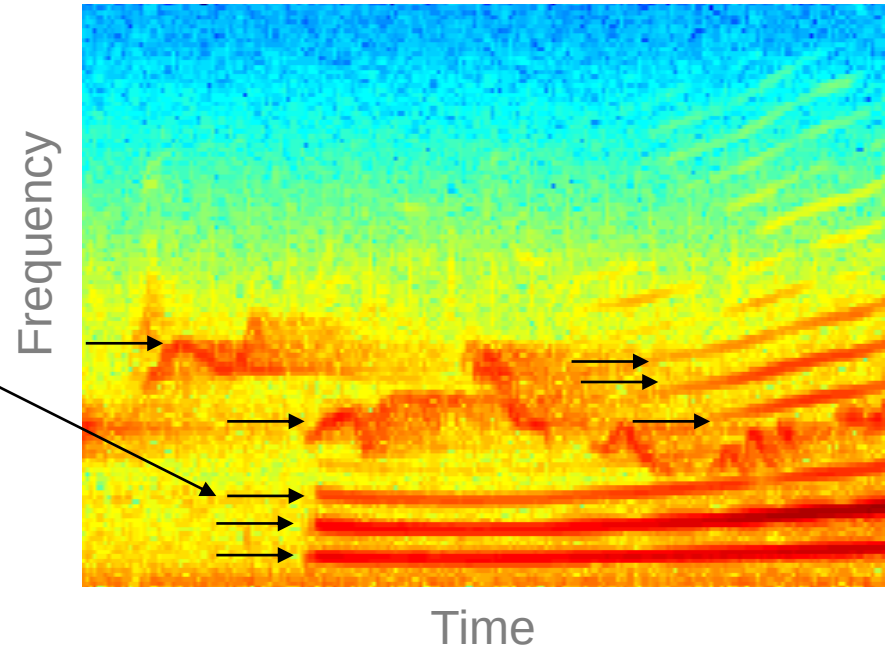
- No need for recordings
- Very simple to implement

Cons:

- Ideal exponential decay doesn't sound realistic
- Most sounds' transients have a non-modal component
  - Generated sounds are too "clean"

Spectral Modeling Synthesis

- Model spectral content rather than physics
- [Serra and Smith 1990]:
  - Sinusoidal partials
    - Amplitude envelope + frequency
  - Noise
    - Spectral envelope
- Good:
  - Quite general and accurate
- Bad:
  - expensive to compute

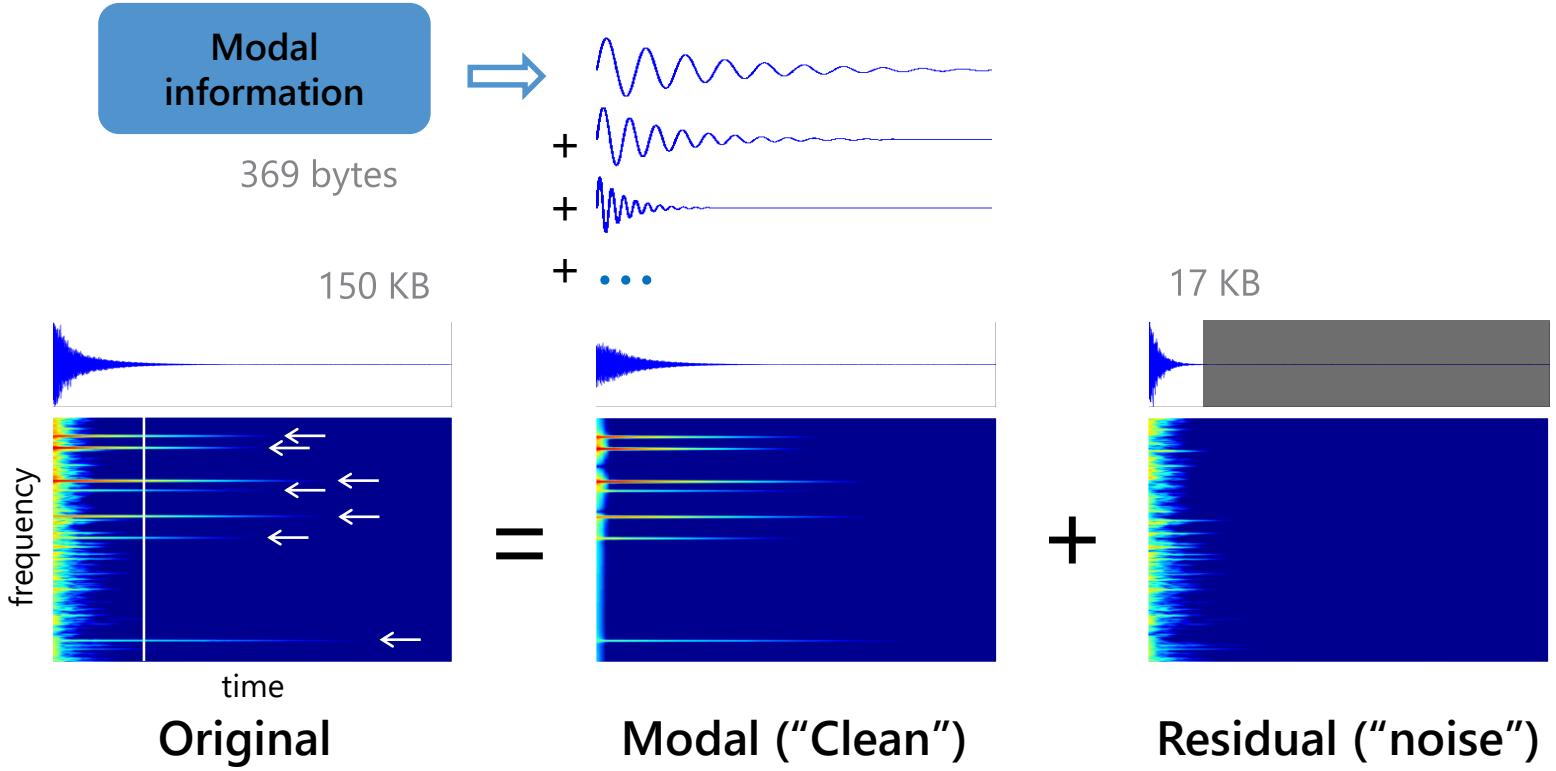


Our approach: middle ground

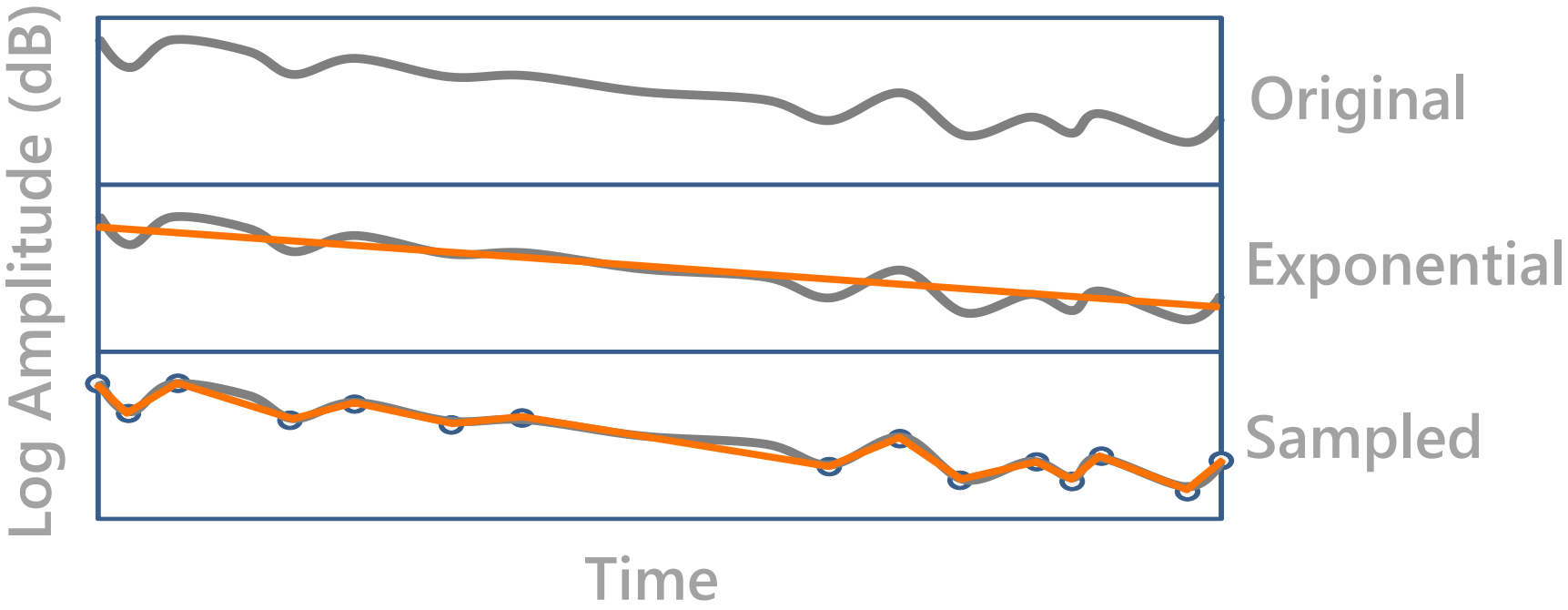
- Arbitrarily decaying sinusoids + residual waveform

- Much more realistic-sounding than forward synthesis
- Faster to compute than spectral modeling synthesis

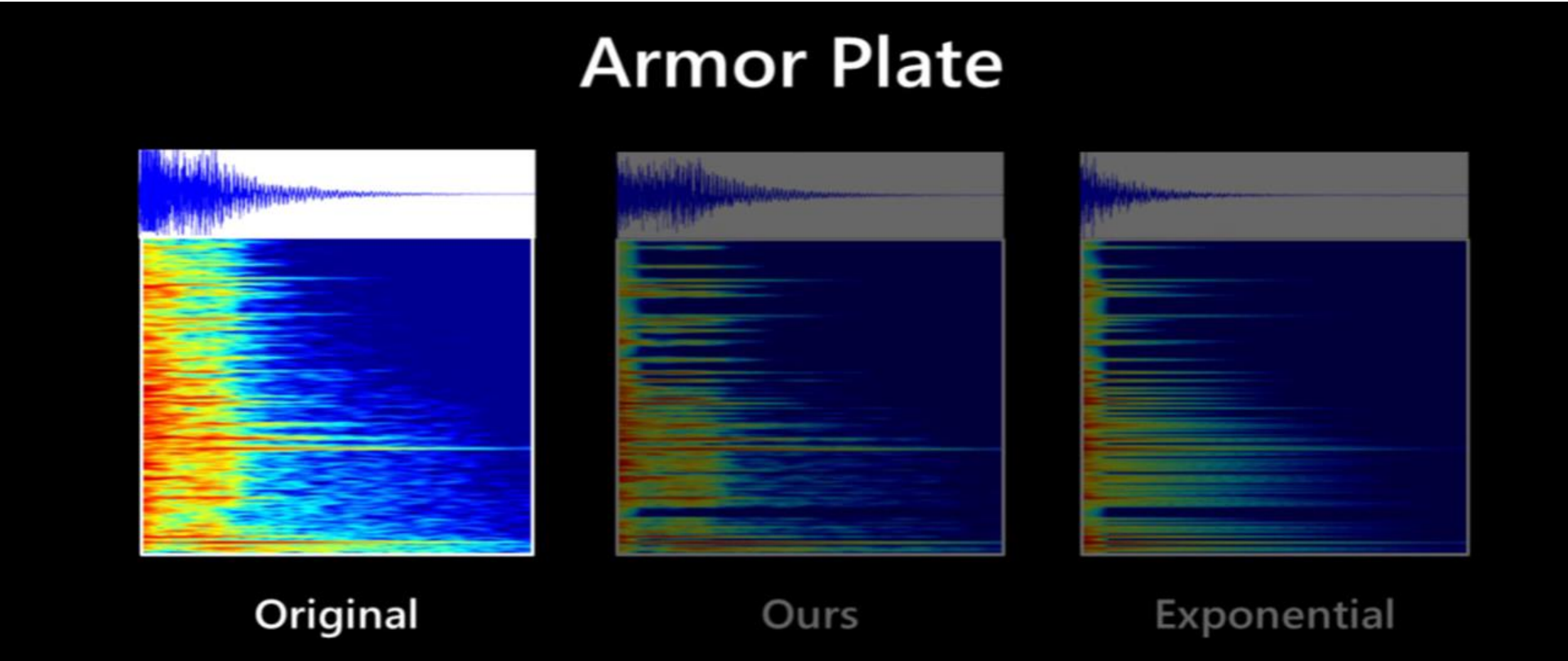
Our approach: details



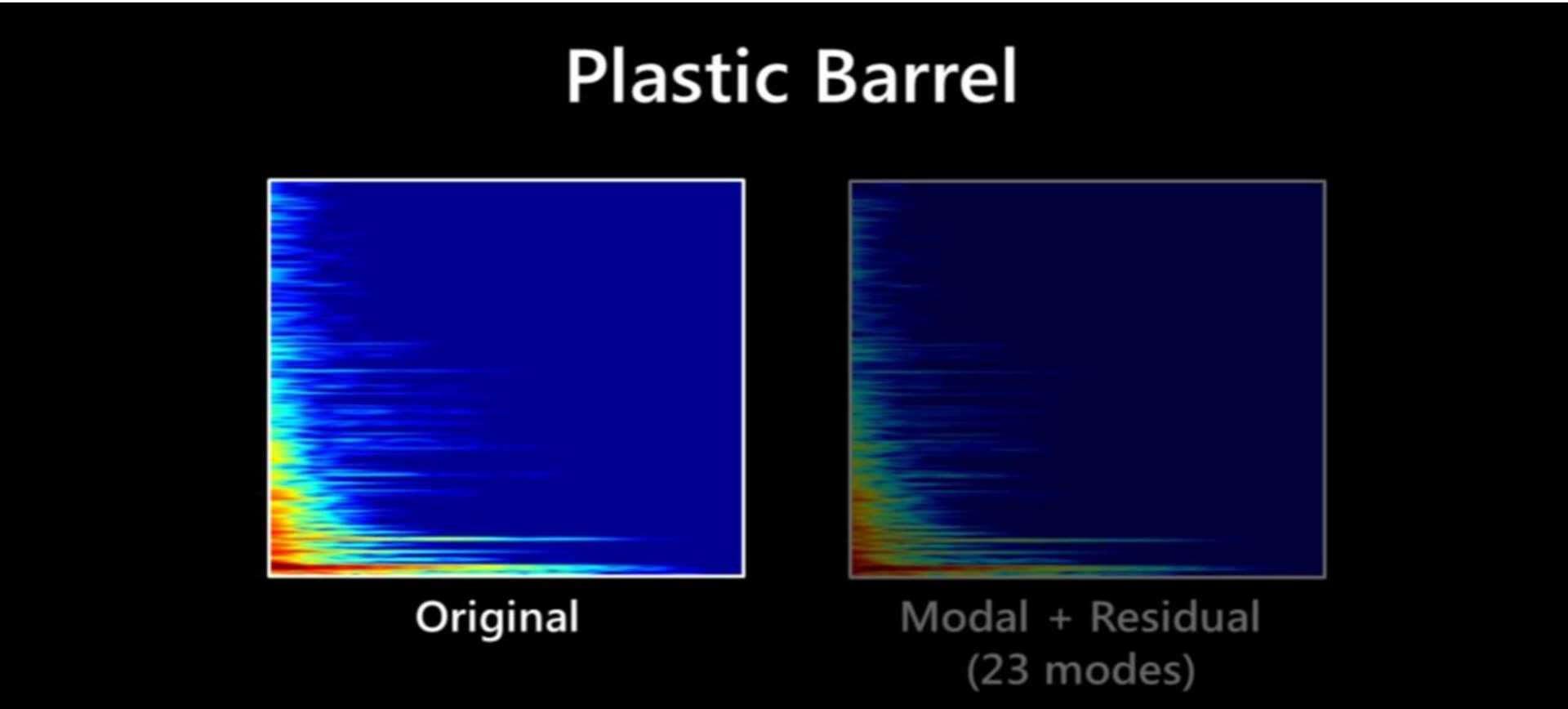
Amplitude Envelopes



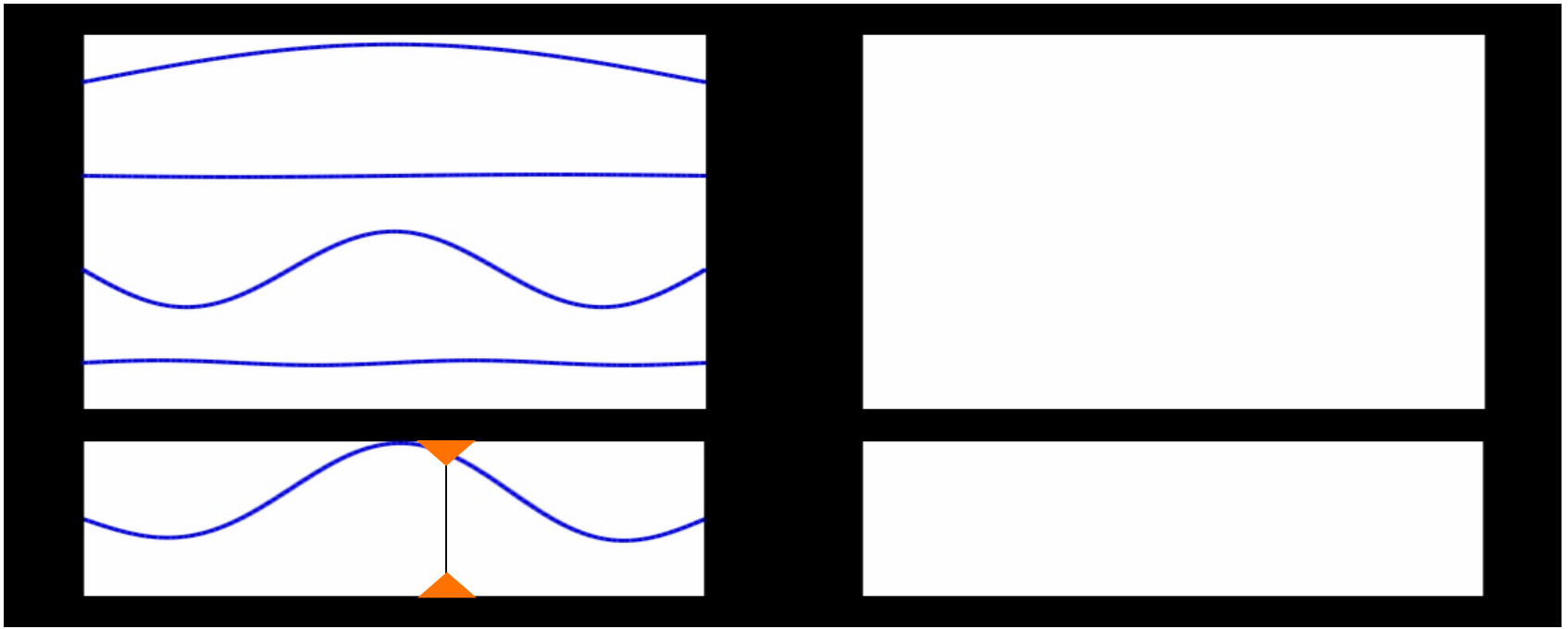
Amplitude Envelopes vs. exponential (modal only)



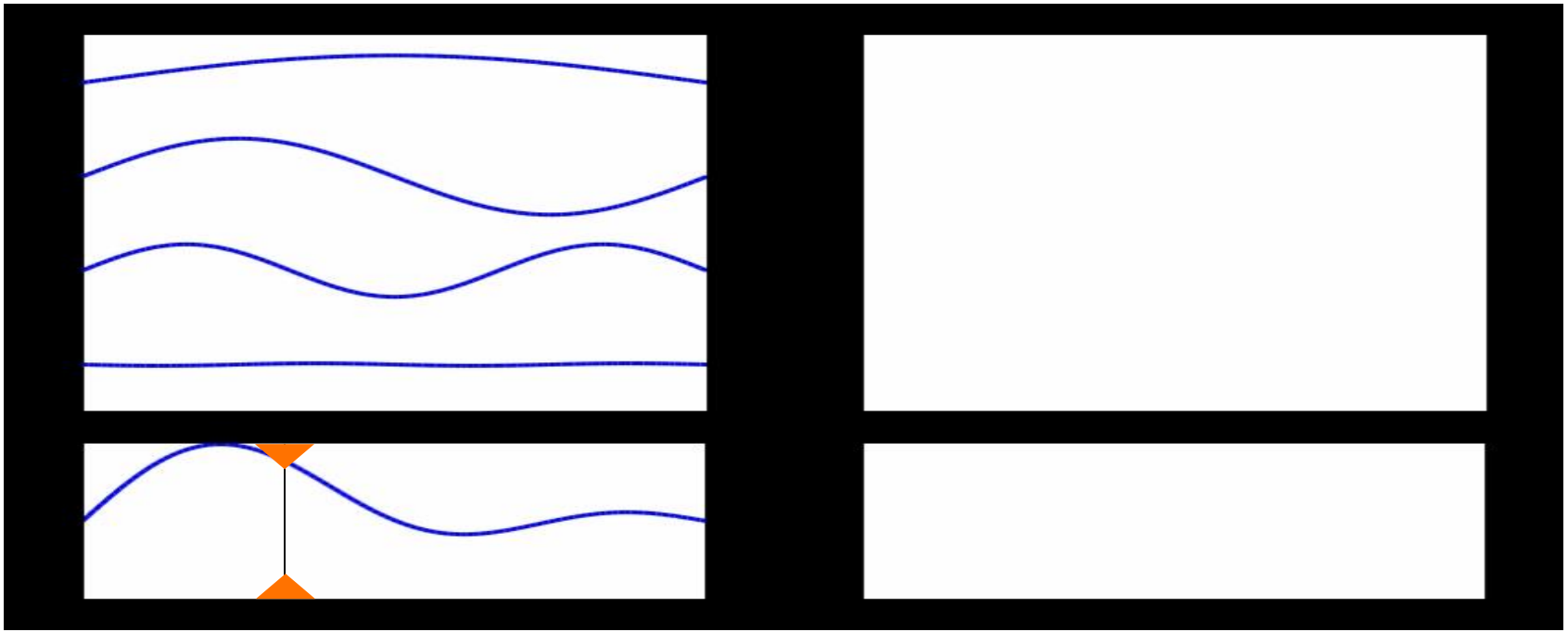
Original vs. synthesized sound



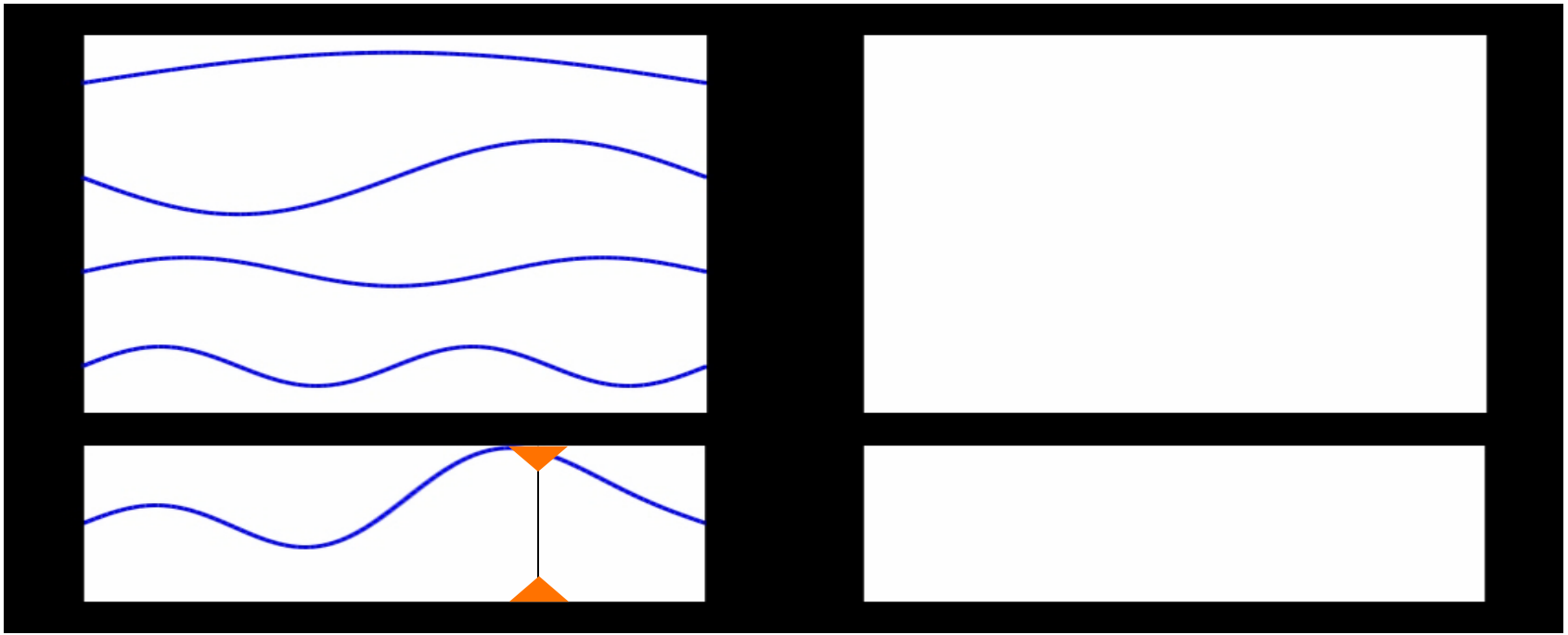
Variation



Variation

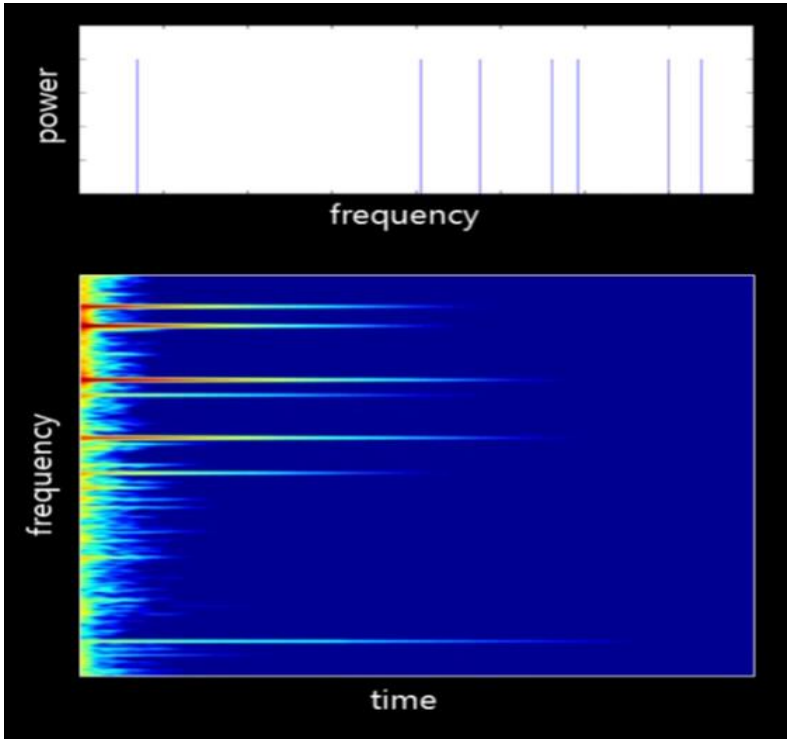


Variation



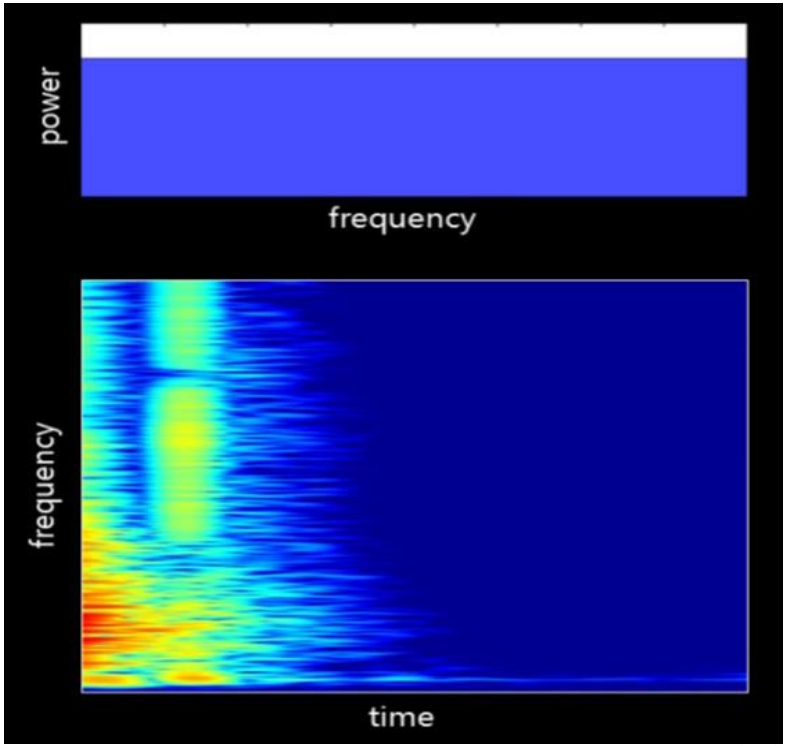
Variation

- Exact variation: mode shapes to determine mode amplitudes
  - requires a lot of memory
- Simple yet effective: Randomize mode amplitudes
  - Degree of variation: artistic control



Variation for Non-modal Sounds

- Attenuate regions of spectrum
- Cascade several dip filters (biquad)
- Randomize parameters
  - location (center frequency)
  - width (Q factor)
  - depth (gain)
- Provides variation from a single clip

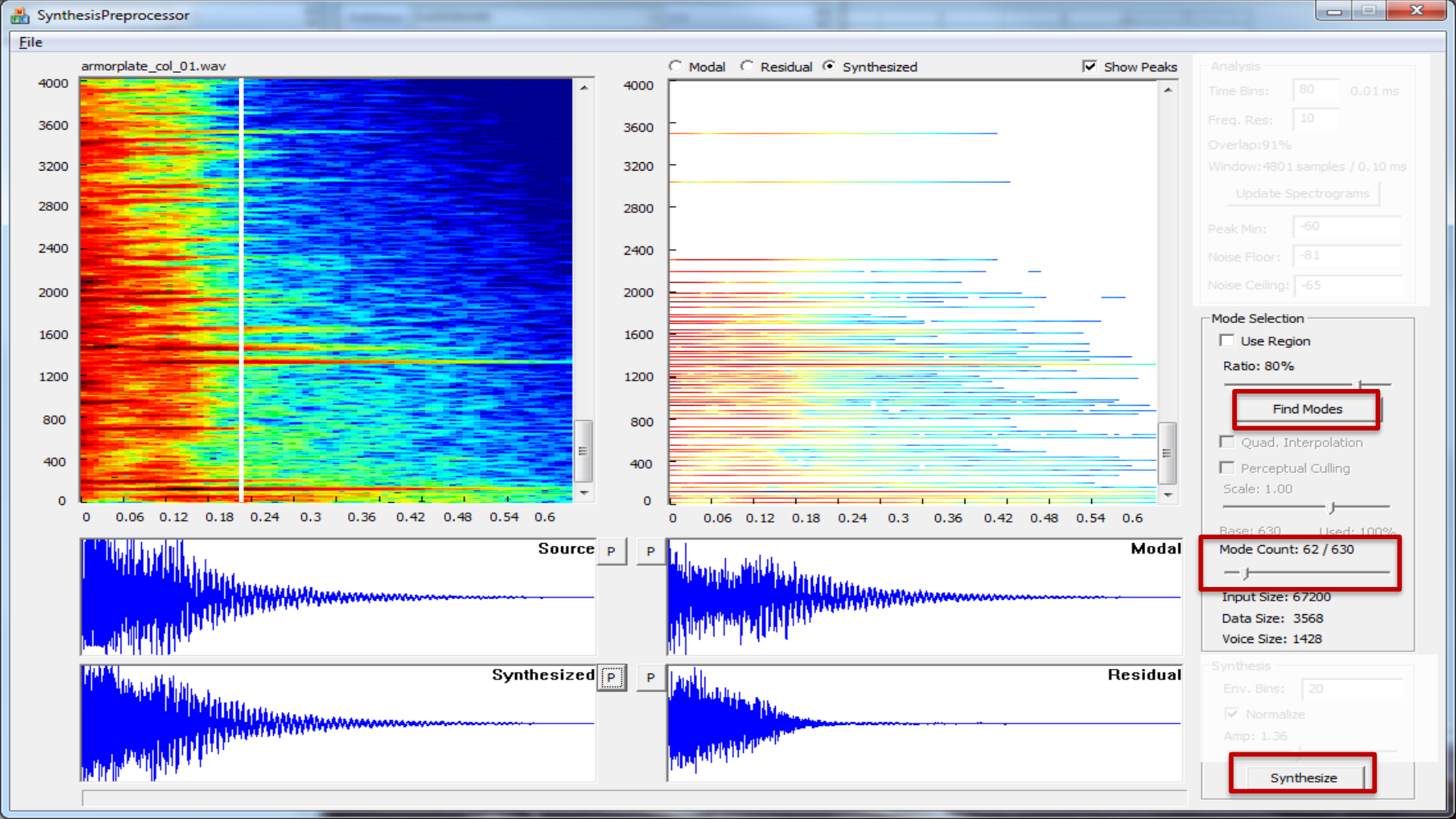


Variation Examples

Name	Variation
Plastic Barrel	Modal
Brass Bell	Modal
Road Cone	Modal
Wooden Stick	Modal
Door	Filter
Ground	Filter
Ice	Filter
Plastic Ball	Filter

Implementation

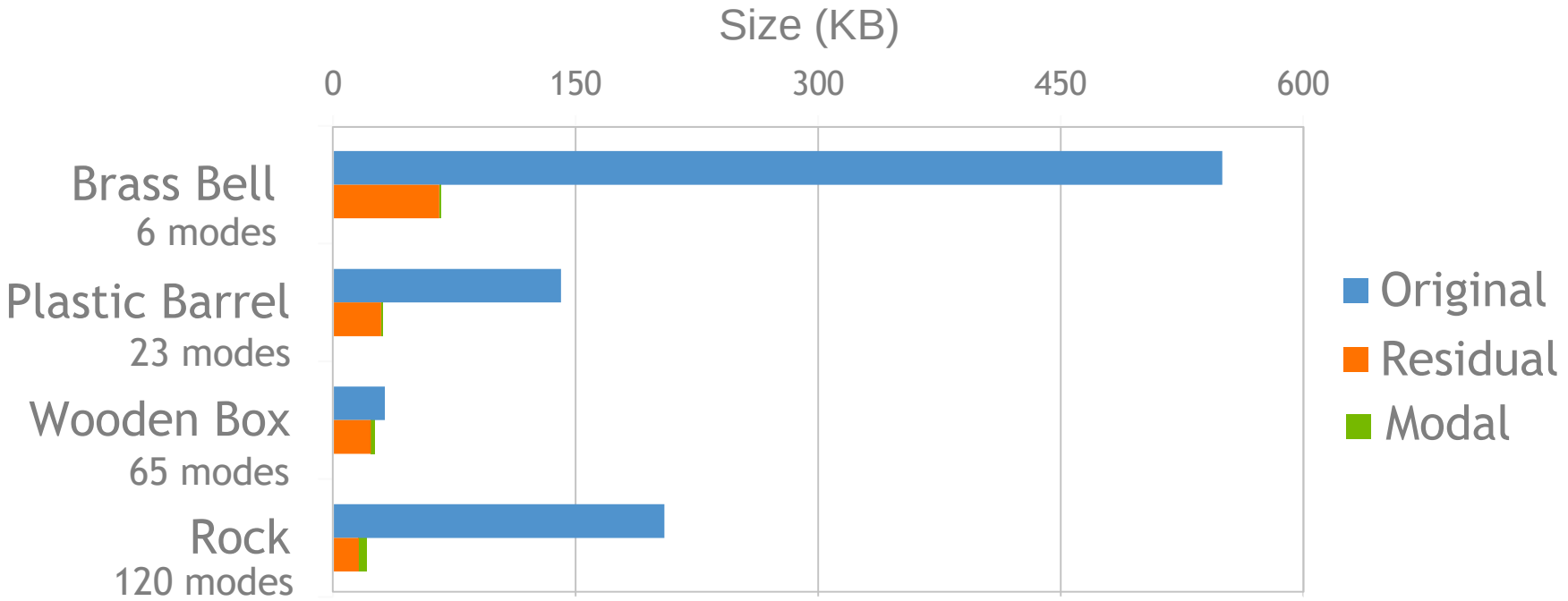
- Xbox360: 3 PowerPC cores @ 3.0 GHz
- Stand-alone analysis tool + plugins for Wwise


SynthesisPreprocessor
File
armorplate_col_01.wav
Modal
Residual
Synthesized
Show Peaks
Analysis
Time Bins: 80 0.01 ms
Freq. Res: 10
Overlap:91%
Window:4801 samples / 0.10 ms
Update Spectrograms
Peak Min: -60
Noise Floor: -81
Noise Ceiling: -65
Mode Selection
Use Region
Ratio: 80%
Find Modes
Quad. Interpolation
Perceptual Culling
Scale: 1.00
Base: 630 Used: 100%
Mode Count: 62 / 630
Input Size: 67200
Data Size: 3568
Voice Size: 1428
Source
Modal
Synthesized
Residual
Synthesis
Env. Bins: 20
Normalize
Amp: 1.36
Synthesize

Improving speed

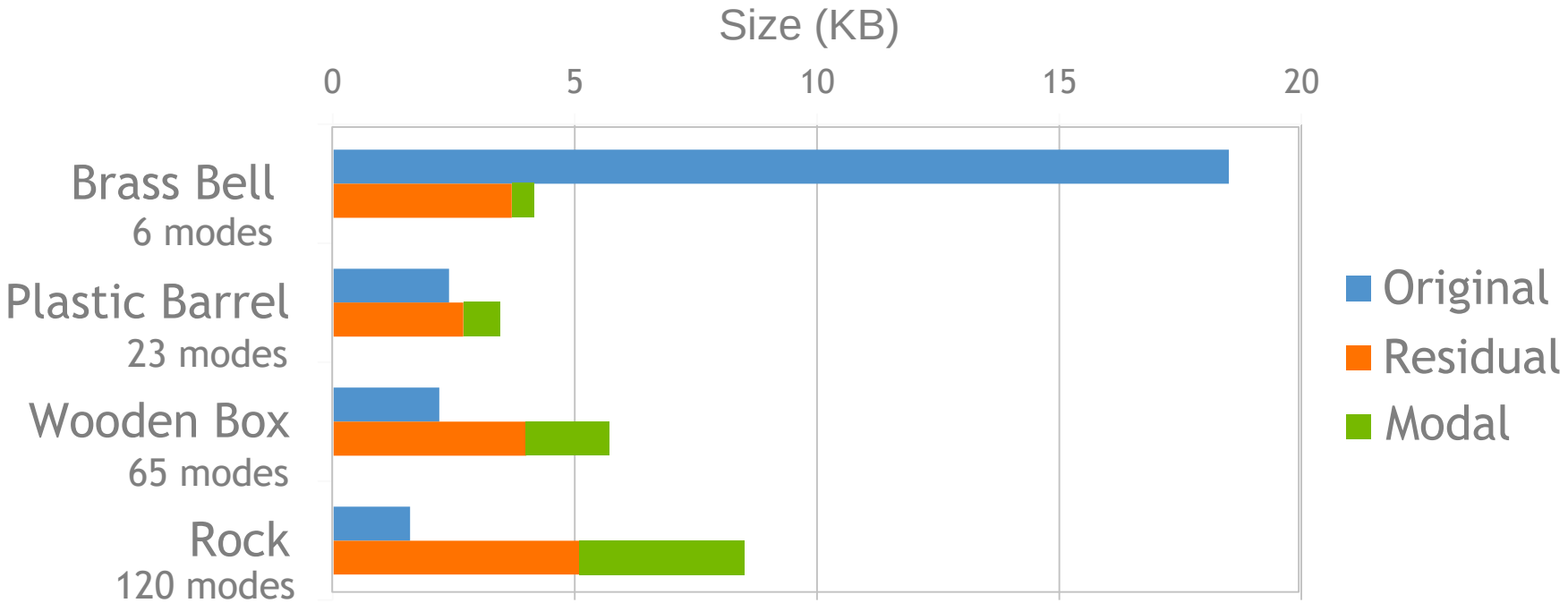
- Computation is **highly** parallel
- Currently in time domain
  - frequency domain would be much faster
  - hard to integrate with middleware like Wwise currently
- Inner loops use vector intrinsics
  - hand-crafted assembly would be 30-40% faster

Memory Usage (1)



Without XMA compression

Memory Usage (2)



Compressed with XMA2 at maximum setting

Modal Synthesis

- single input clip
- no variation

Modal Synthesis

- single input clip
- with variation

Quality Scaling

- Fixed mode budget
  - fixed memory overhead
  - fixed CPU usage
- Prioritize modes by how much they contribute
- Remove low priority modes

Quality Scaling: Reference

- high mode quota (5000)

Quality Scaling

- low mode quota (1000)
- 5x savings in compute

Variation Filter

- single clip


Single clip
(no variation)

Summary

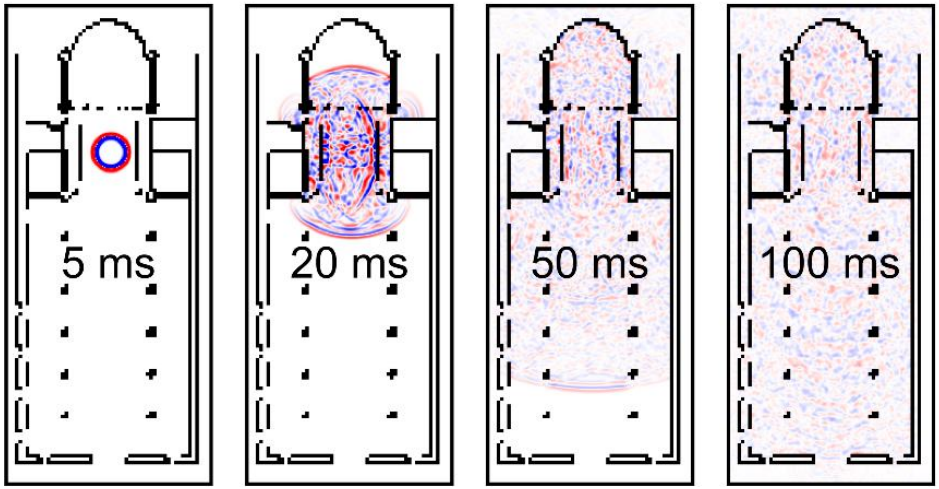
- Sound synthesis for high quality impact sounds with variation in Crackdown 2
  - more compute
  - less memory
  - infinite variety
- Powerful physically-based representation
  - rolling/sliding could potentially be modeled
  - more research needed

Overview

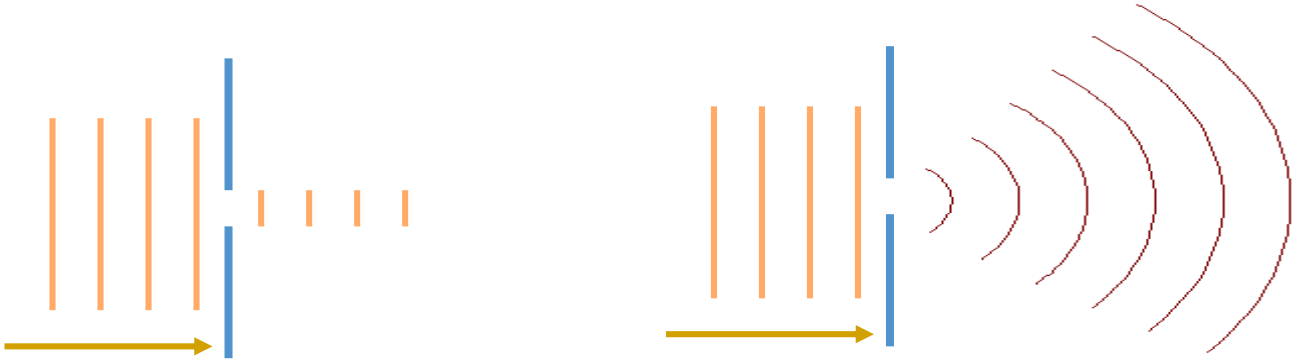
- Physics of sound
- Sound synthesis in Crackdown 2
- Wave acoustics
- Conclusion

Sound Propagation

- Essential for immersion in acoustic spaces
- Physically complex, perceivable effects



Diffraction

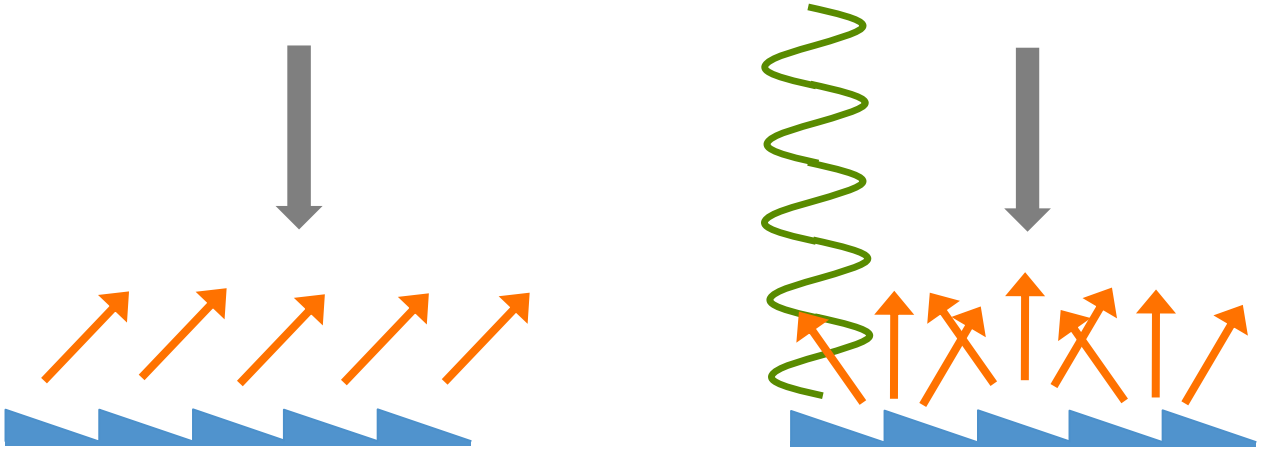


Light

Sound

- Sound does NOT propagate in rays like light, it bends
- So, you often hear what you don’t see
- Low-frequencies bend more: occluders act as low-pass filters

Scattering



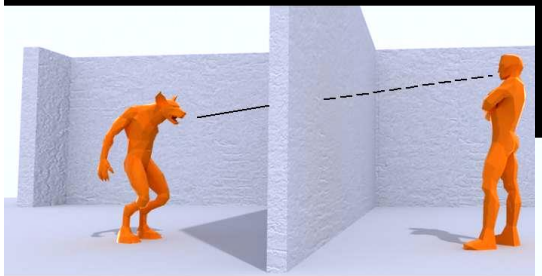
Light

Sound

- Geometric details cause scattering: A furnished room sounds very different from an empty one

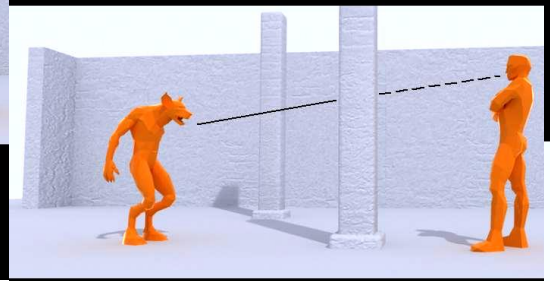
Games today

Occlusions



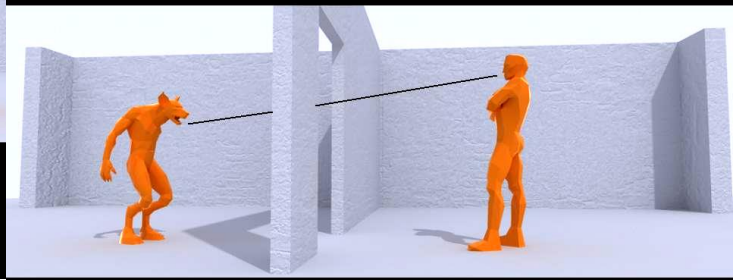
Direct path muffled
Reflections muffled

Obstructions



Direct path muffled
Reflections clear

Exclusions



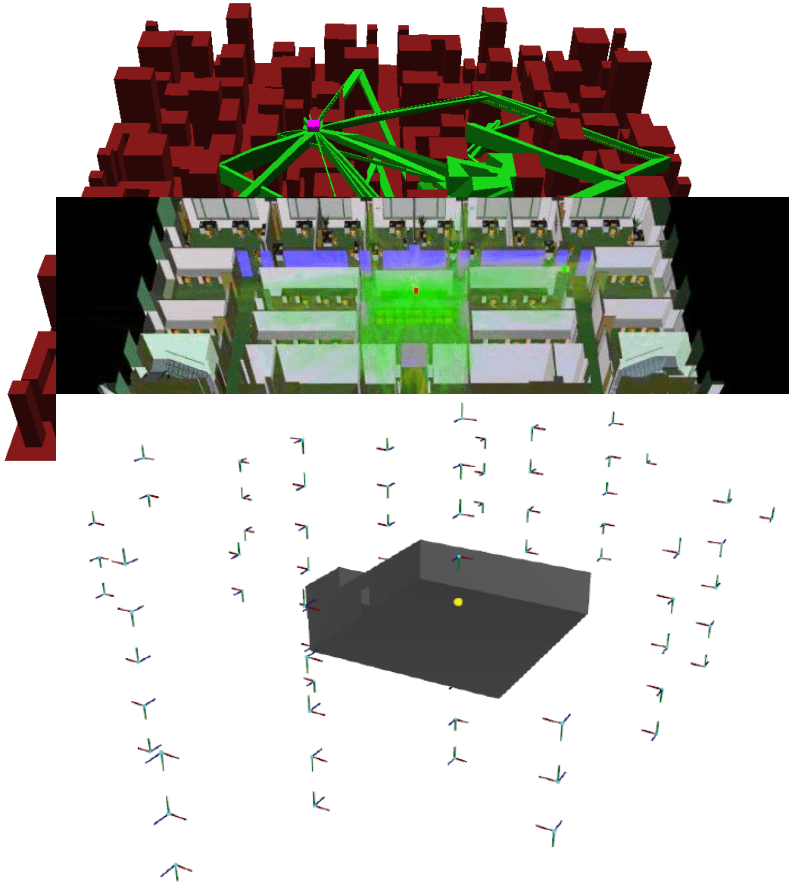
Direct path clear
Reflections muffled

Hand-tuned filters: difficult & tedious

Simulated propagation → automatic, detailed, scene-dependent acoustics

Ray-based methods

- Beam-tracing
  [Funkhouser et. al. 2004]
- Ray frustum-tracing
  [Chandak et. al., 2008]
- Image-source method
  [Tsingos, 2009]
- Problems: diffraction/scattering, high-order reflections challenging



Our Approach

- Pre-computed wave simulation
  - attenuation behind obstructions, low-pass filtering
  - scattering from complex surfaces
  - realistic, scene-dependent reverb
  - works with “triangle-soup” scenes
- Pre-compute & render
  - scene geometry not needed at runtime

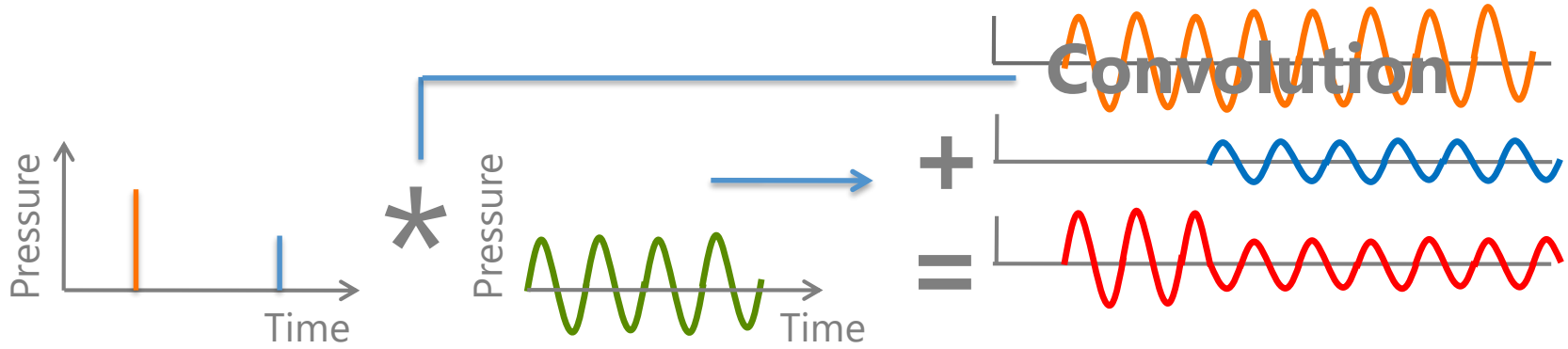
Comparison with Half Life 2™

- “Train Station” scene
- Our engine vs. game’s default sound engine

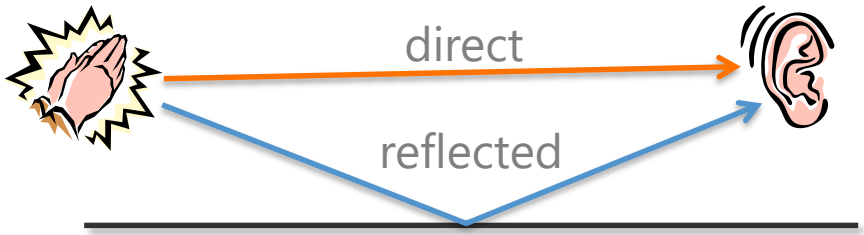

Half-Life 2 engine

Our engine

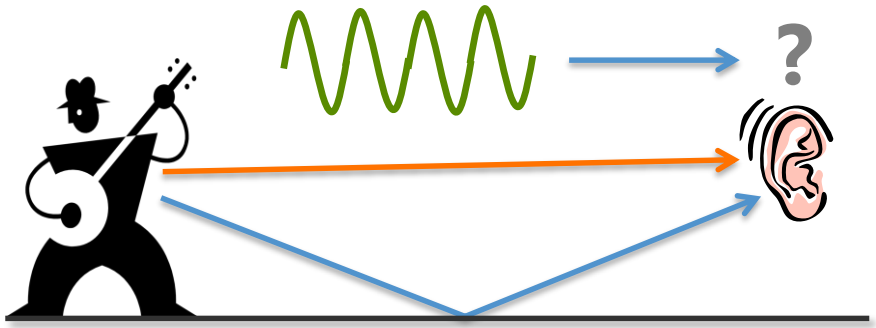
Pre-compute & Render



Impulse Response

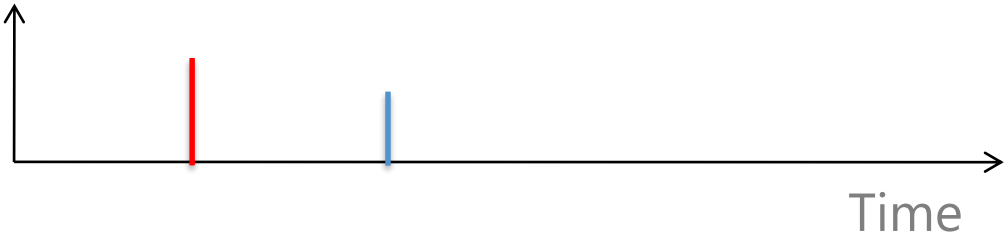


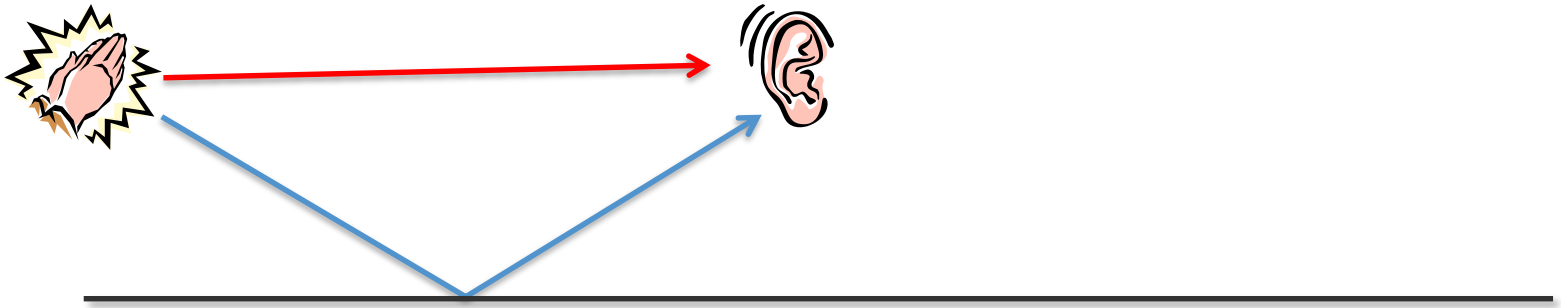
Precompute



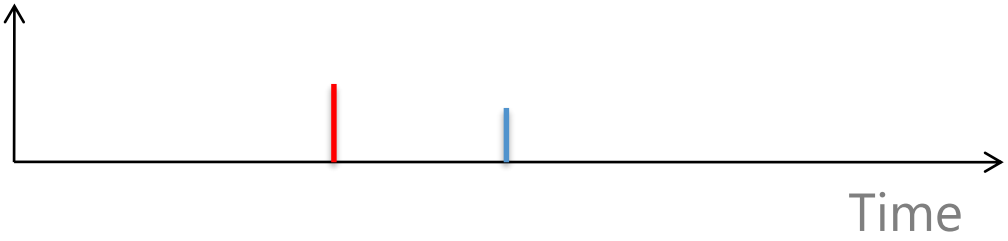
Render

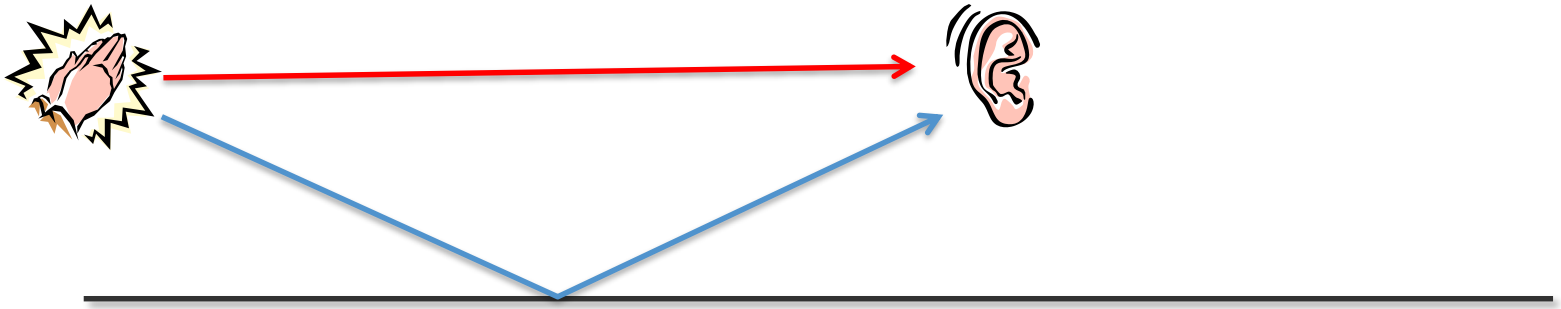
Moving Source & Listener



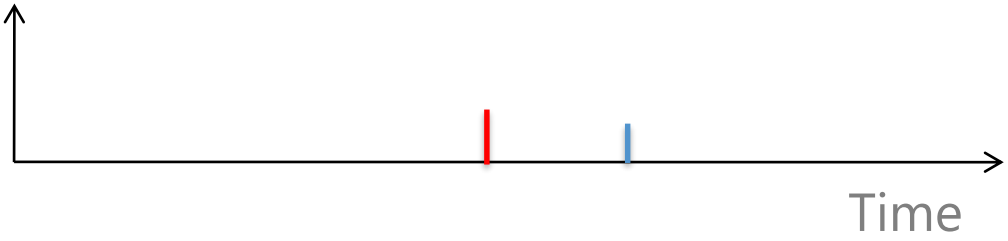


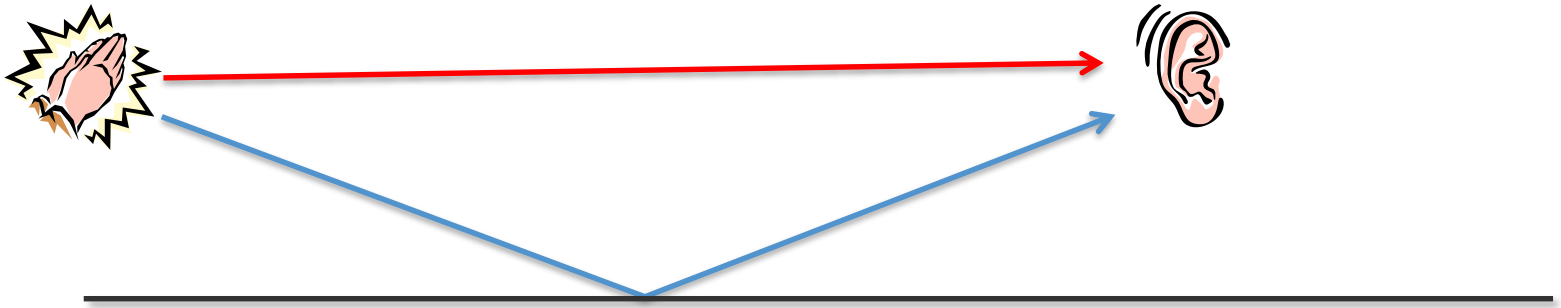
Moving Source & Listener



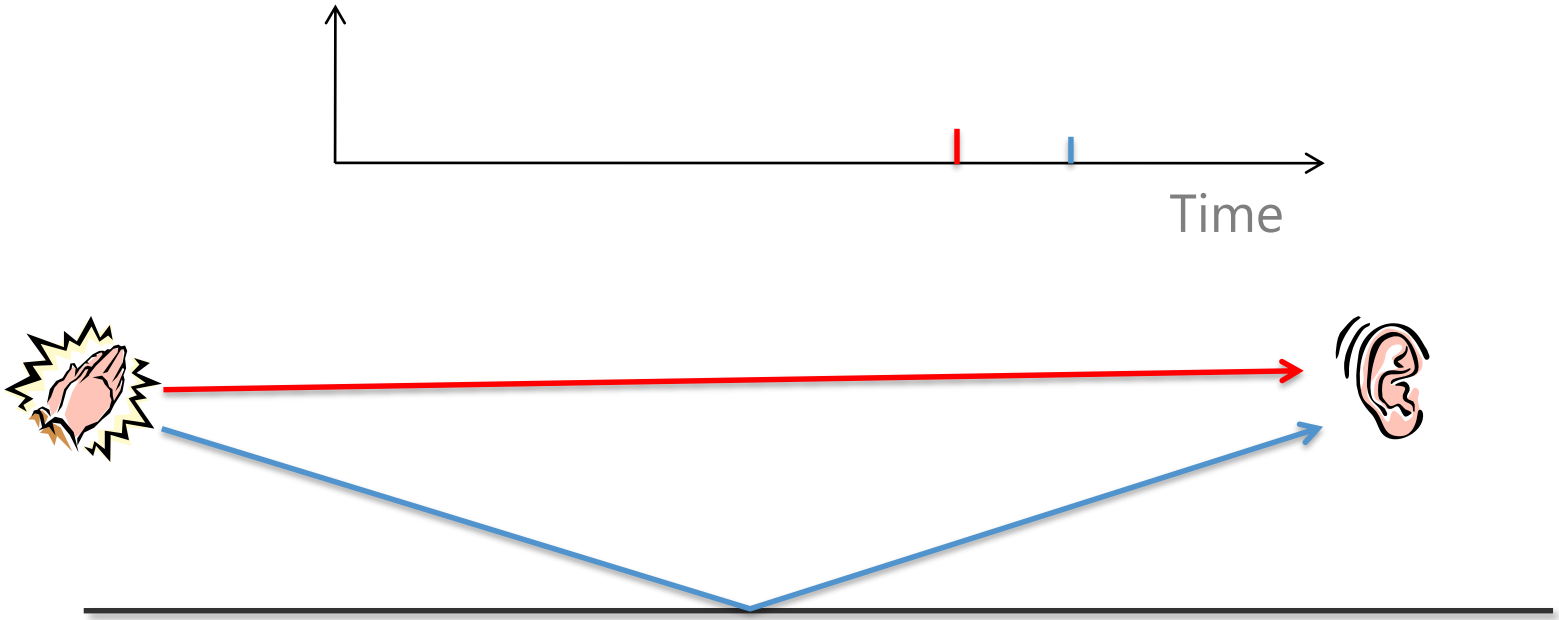


Moving Source & Listener

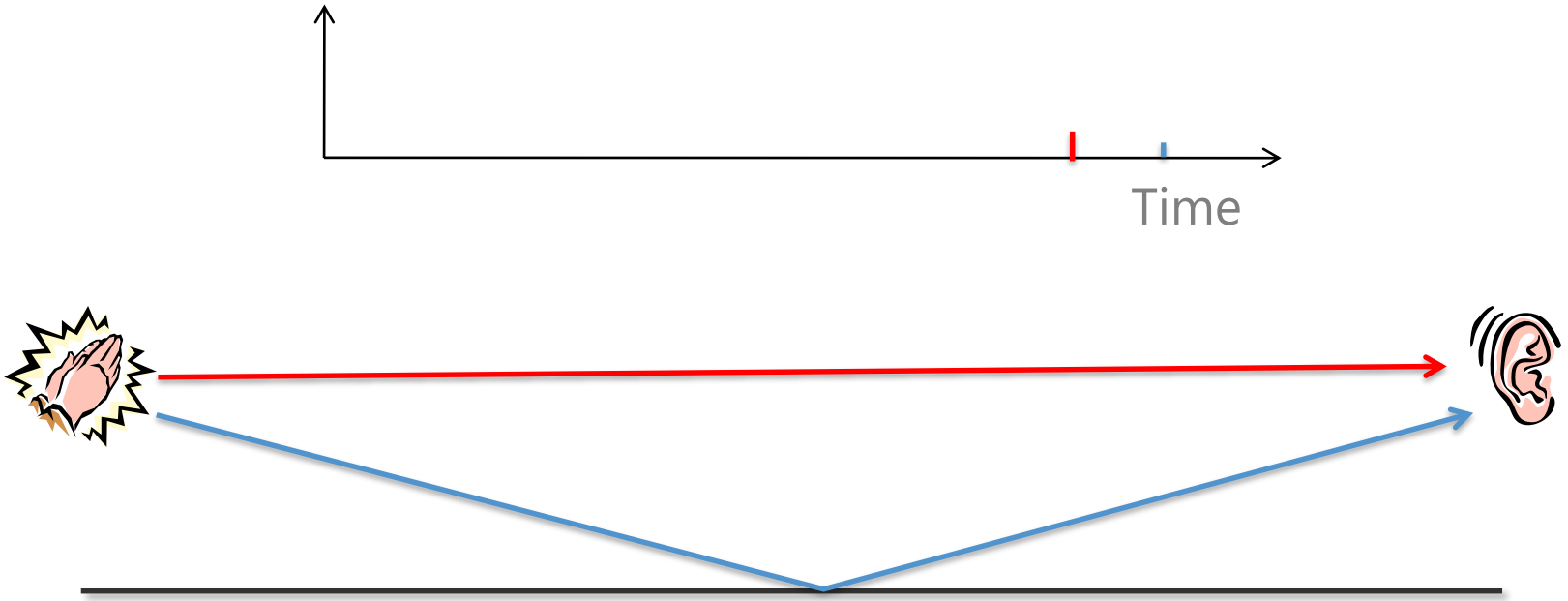




Moving Source & Listener



Moving Source & Listener



Main Challenge

- Acoustic responses vary spatially
- **7D** sampling space
  - source (3D) x listener (3D) x time (1D)
- Can reduce to 6D
  - Restrict listener to 2.5D surface
- Brute-force sampling still infeasible

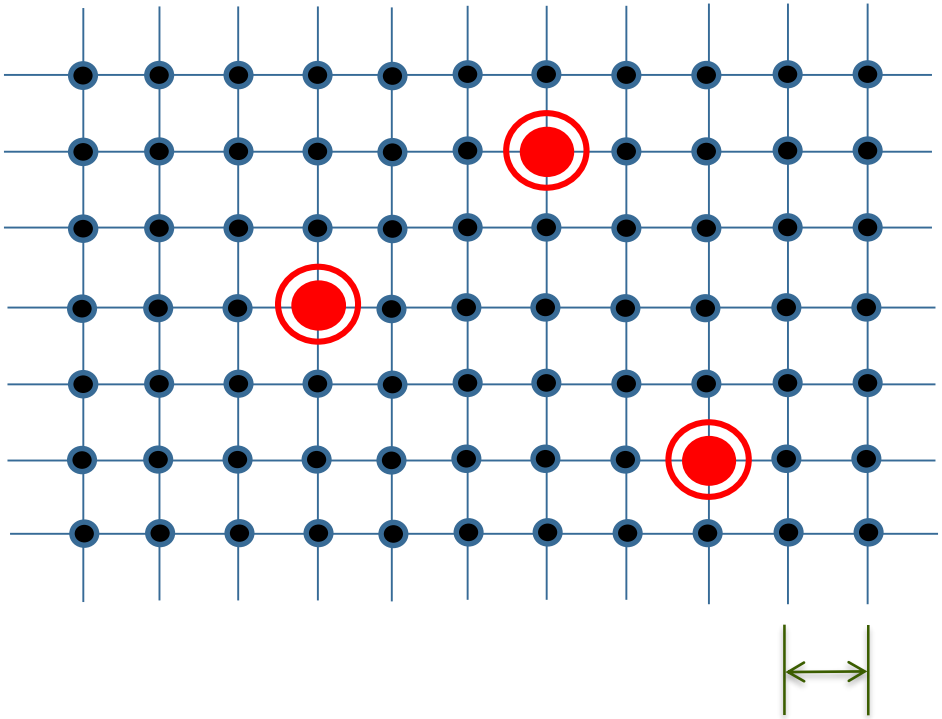
Brute-force sampling

- Scene: Half-Life 2's Citadel (28m x 60m x 32m)
- Simulation grid (12 cm)
- Direct storage:
  200,000 GB

Main ideas

- Compact representation for acoustic impulse responses
  - works well with wave simulations
- Spatial interpolation allowing coarse (~1m) grid
- Real-time wave effects
  - automatic correspondence to scene geometry

Our approach: Overview

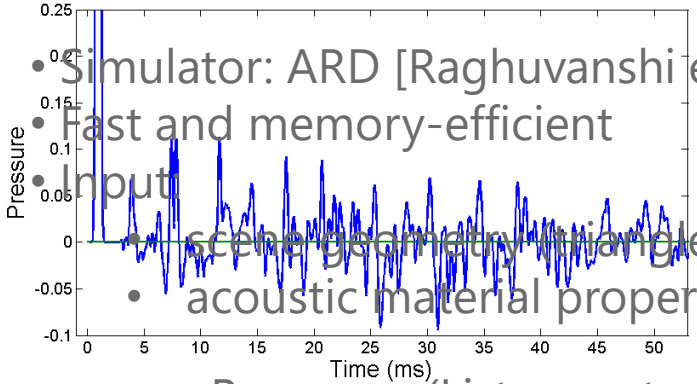


~ 1-2 meters

Source locations

Listener locations

Wave Simulation

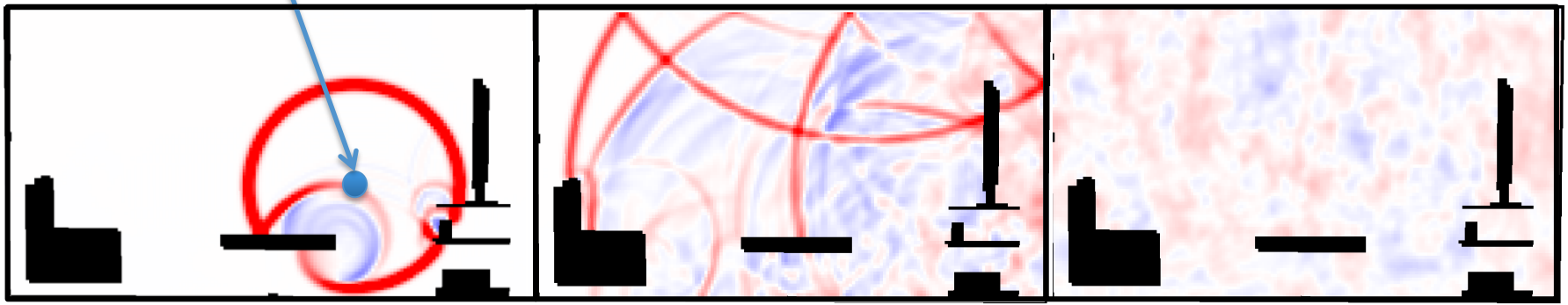


- Simulator: ARD [Raghuvanshi et. al., 2009]
- Fast and memory-efficient
- Input:
  - scene geometry (triangle mesh)
  - acoustic material properties

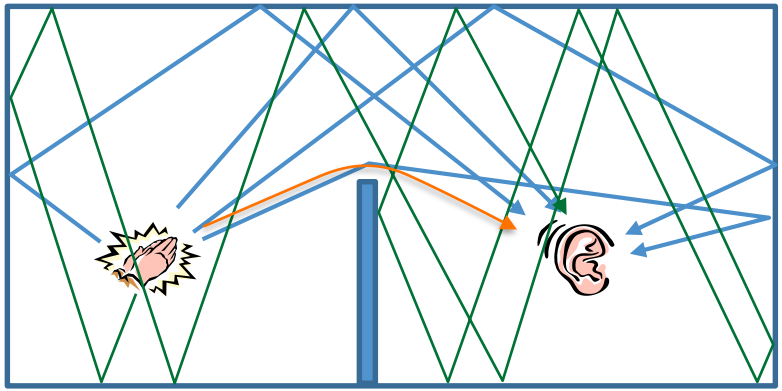
Encode this compactly

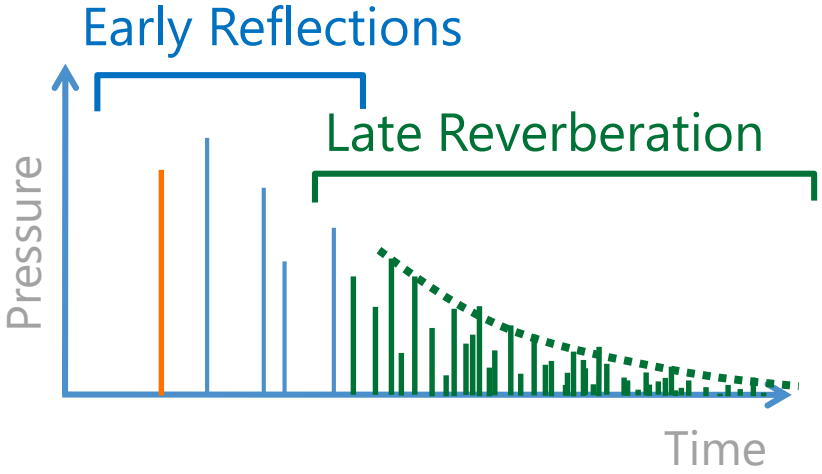
Source pulse

Response (Listener at source location)



Psycho-acoustics





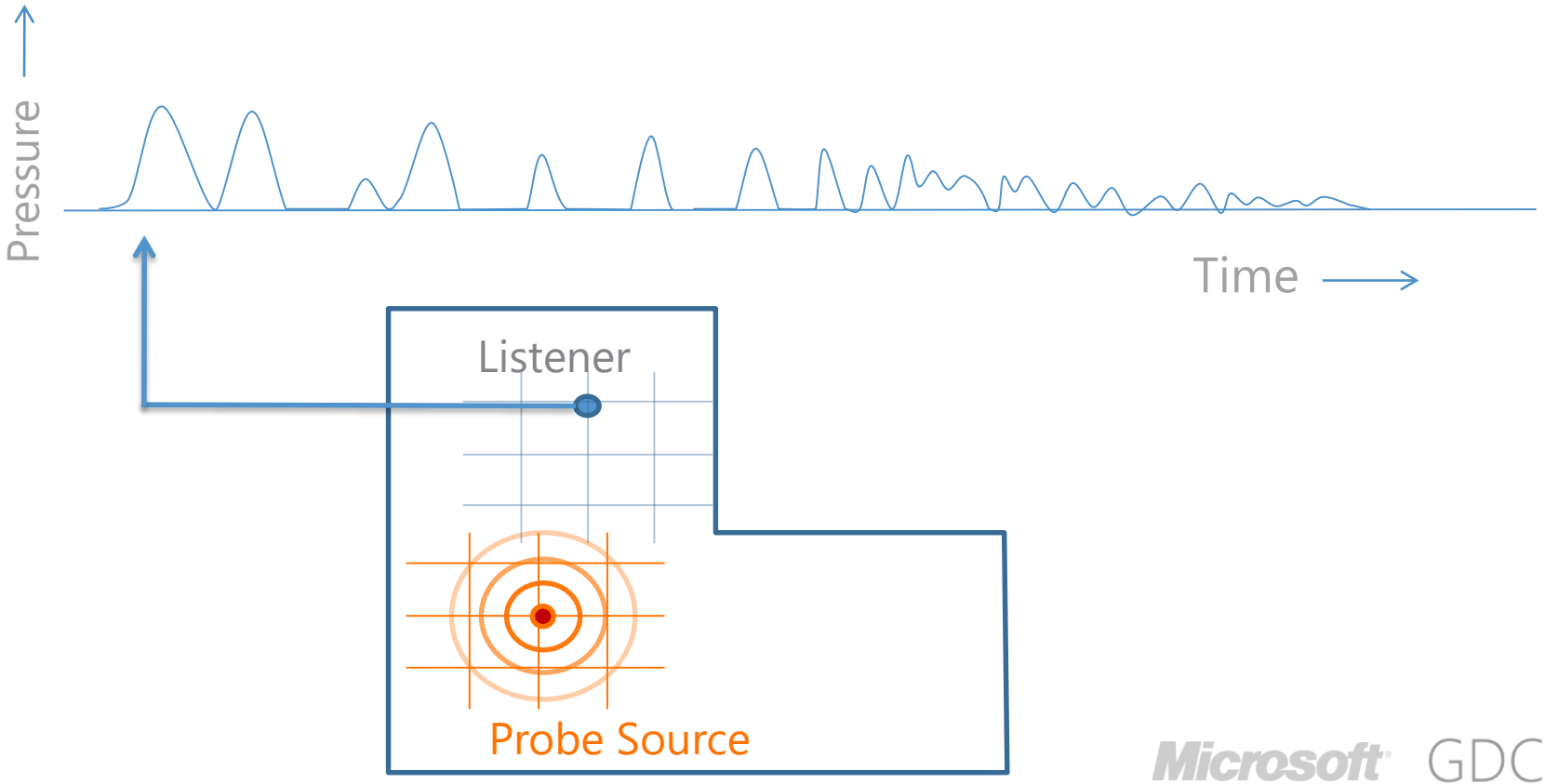
- Direct Sound: sense of direction
- Early Reflections: loudness, timbre; spatial variation; ~100 ms
- Late Reverberation: decay envelope; no spatial variation; few seconds

Reference: “Room Acoustics” by Heinrich Kuttruff

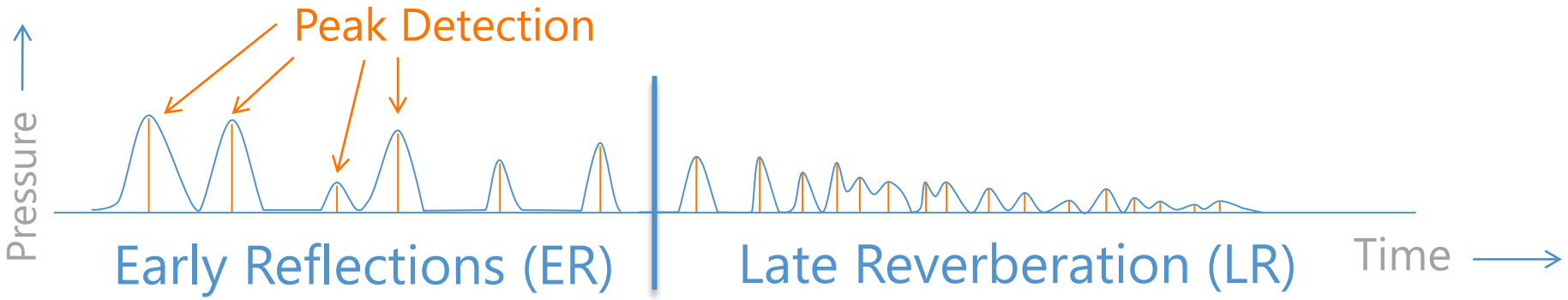
Demo

- Perceptual effect
  - Only Direct sound
  - Direct + Early Reflections
  - Direct + Early Reflections + Late Reverberation

Technique: Wave Simulation

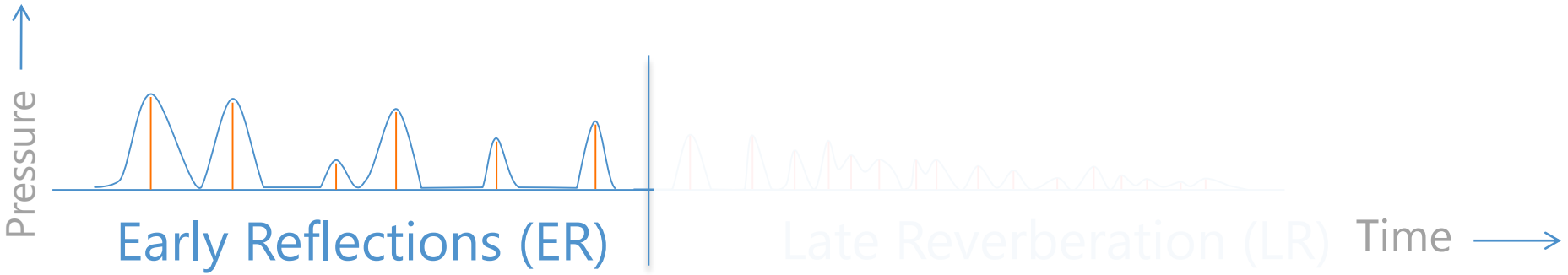


Early Reflections / Late Reverb



- Automatic separation: echo density (500 peaks/sec)
- **single** late reverb filter per room: room-dependent reverb
- early reflection sampled on a grid within room

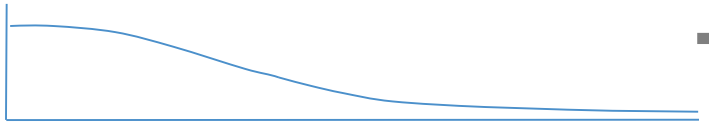
ER Representation



Store

Peak times and amplitudes

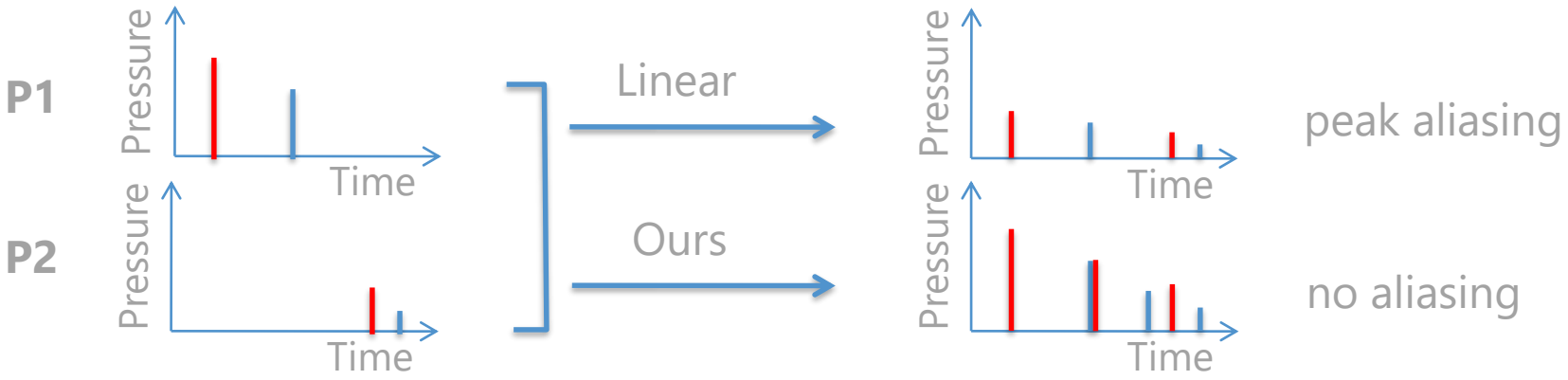
captures reflection/diffraction strengths and delays

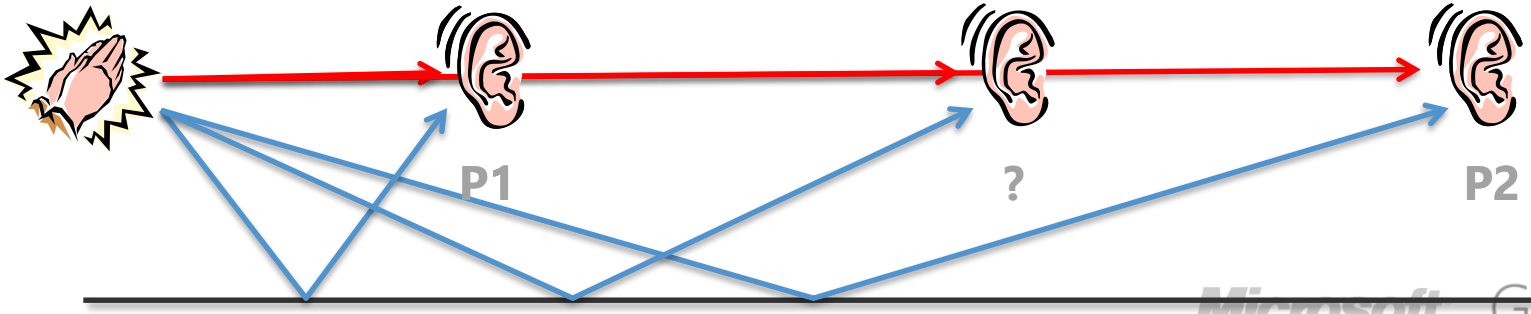


Frequency trend

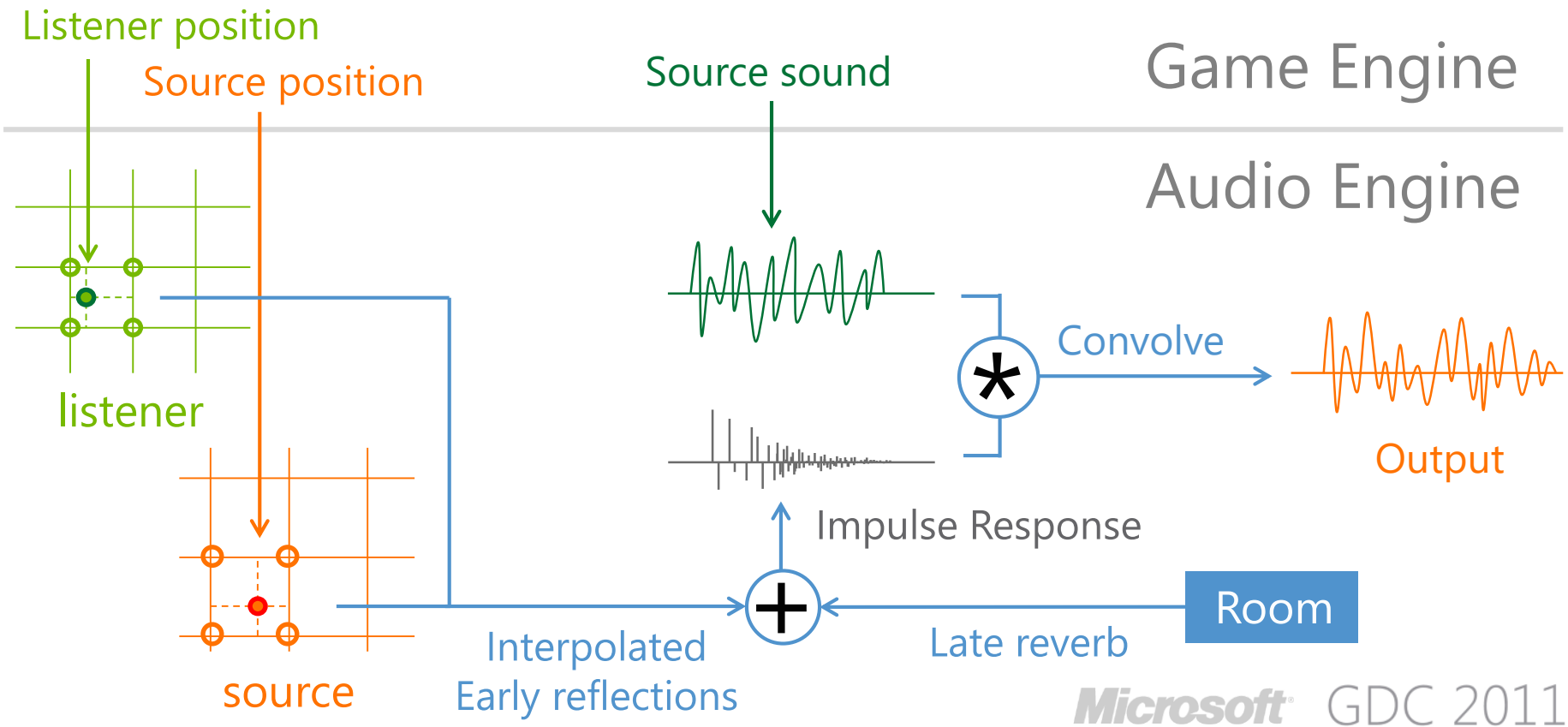
captures spectral effects (low-pass filtering etc.)

Runtime: Spatial Interpolation





Runtime: Audio Engine

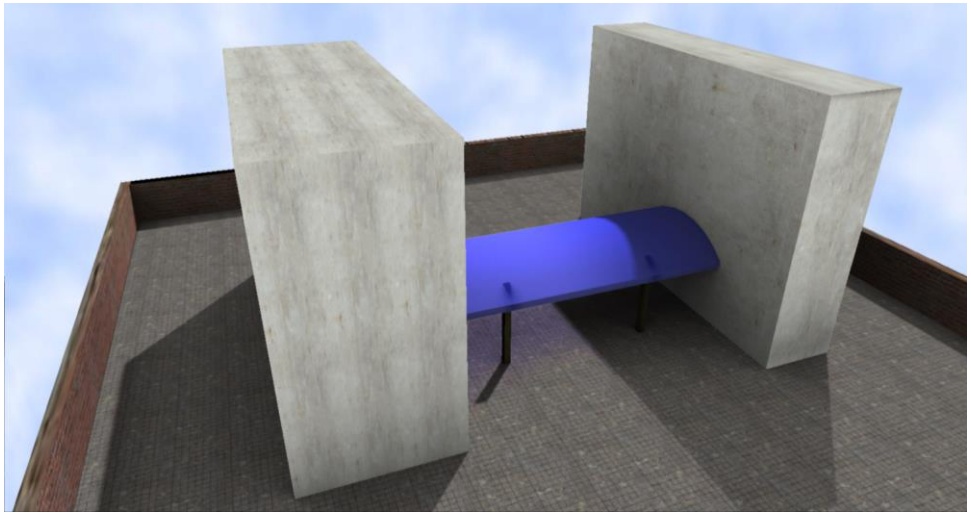


Results: Citadel Walkthrough

- Game environment from Half Life 2™
- Size: 60m x 28m x 22m


D3D9 22.59 fps Vsync off (1600x900), X8R8G
(HAL (hw vp): NVIDIA GeForce GTX 295

Results: Walkway



- Size: 19m x 19m x 8m
- Realistic attenuation behind walls


D3D9 32.00 fps Vsync off (1366x768), X8R8G8B8 (D24X8) (4x Multisample)
HAL (hw vp): NVIDIA GeForce 9600M GT

Results: Walkway

- Sound focusing
  - Loudness increases beneath concave reflector


D3D9 116.18 fps Vsync off (1366x768), X8R8G8B8 (D24X8) (4x Multisample)
HAL (hw vp): NVIDIA GeForce 9600M GT

Results: Living Room

- Empty vs. furnished living room
- Scattering off furnishings changes acoustics


D3D9 236.61 fps Vsync off (1366x768), X8R8G8B8 (D24X8) (4x Multisample)
HAL (hw vp): NVIDIA GeForce 9600M GT


D3D9 140.61 fps Vsync off (1366x768), X8R8G8B8 (D24X8) (4x Multisample)
HAL (hw vp): NVIDIA GeForce 9600M GT

Integration with Half Life 2™

- Train Station scene: 36m x 83m x 32m


HEALTH
100
SUIT
0
AMMO
6
12

Cost

- Run-time: ~5% of a core per source
  - Reduce further: partitioned convolution used in Wwise
- Memory: <100 MB – current work to reduce to <10 MB
- Pre-compute: few minutes per source location, few hours per scene

- Reference processor: 2.8GHz Quad-core Intel Xeon

Summary

- First real-time wave-based sound propagation engine
  - Automatic diffraction, occlusion/obstruction, late reverberation, focusing, in complex 3D scenes
- Moving sources and listener

Future Work

- Need further 10x reduction in memory: ~5 MB footprint
  - spatial compression
  - adaptive sampling
- Dynamic geometry could be added
  - Pre-computed frequency-dependent occlusion factors
- Integration with existing audio middleware

Overview

- Physics of sound
- Sound synthesis in Crackdown 2
- Wave acoustics
- Conclusion

Conclusion

- Physically-based impact sounds in Crackdown 2
  - infinite variation
  - saves memory
- Wave acoustics for games
  - automatic scene-dependent roll-off, obstructions, focusing, reverb etc., depending on source/listener location
- Physically-based techniques are a powerful way to deal with game sounds
  - automation for the tedious part of audio design

Thank you!! Questions?

- http://research.microsoft.com/people/nikunjr/
- Papers, demos etc.

- nikunjr@microsoft.com