Interactive Sound Rendering

Sound Synthesis
Numerical Acoustic Simulation
Interactive Sound Propagation
Application Demonstration

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

Overview

- Interactive Sound Synthesis
- Efficient Numerical Acoustic Simulation
- Interactive Sound Propagation
- Conclusion and Future Work

Overview

- Interactive Sound Synthesis
  - Modal Analysis
  - Perceptually-based acceleration techniques
  - Hundreds of sounding objects
  - Sound from image textures
- Efficient Numerical Acoustic Simulation
- Interactive Sound Propagation
- Conclusion and Future Work

Physically-based Sound Synthesis

- Aim: Take object geometry and material as input and produce sound
- Current trend: Recorded sounds
- Problems with recorded sounds:
  - Difficult, expensive or dangerous to record (eg. Explosions)
  - Repetitiveness
  - Complex interactions (impact/rolling)

A typical foley studio*

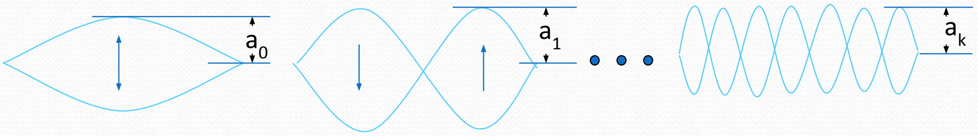
* Image taken from: http://www.marblehead.net/foley/index.html

---

Physical Simulation

- Elastic deformable model
- Typical simulation time-steps must be ~10^{-5} s
- Direct simulation infeasible
- Efficient method: Modal Analysis

Modal Analysis



1st Mode

Frequency = f_0

2nd Mode

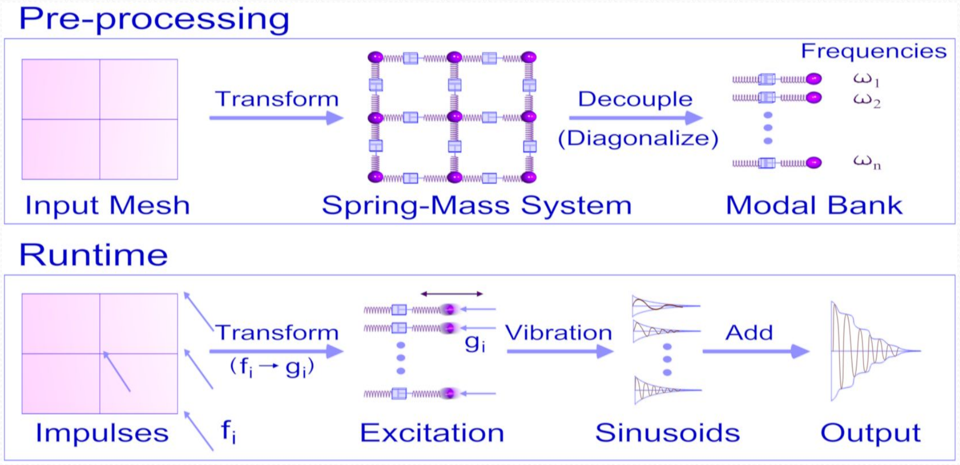
Frequency = $f_1 = 2*f_0$

…Higher modes

Frequency = $f_k = k*f_0$

- Each mode represents a resonant mode of vibration
- Frequency of a mode is fixed
- Applying impulse excites modes of vibration
- Position of impact determines relative amplitude of modes

Overview of Technique



Approach

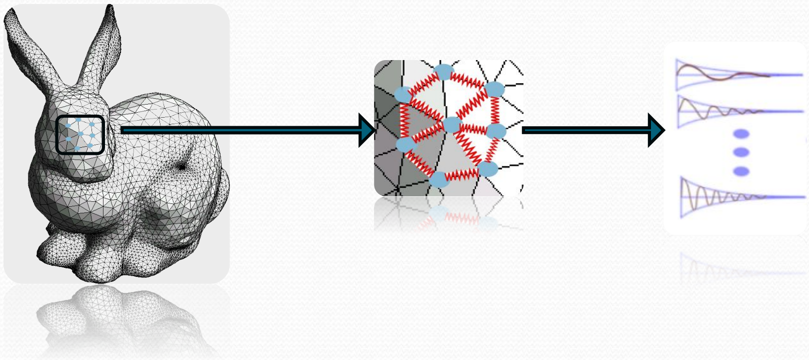
- Simpler model: Spring-mass system
- Fast: Supports hundreds of sounding objects
- Runs in real-time, low CPU utilization (~10%), graceful degradation in quality with limited computation
- Exploit human auditory perception
  - Mode Compression
  - Mode Truncation
  - Quality Scaling

Modal Analysis

- Deformation modeling
  - Vibration of surface generates sound
  - Sound sampling rate: 44100 Hz
  - Impossible to calculate the displacement of the surface at sampling rate
  - Represent the vibration pattern by a bank of damped oscillators (modes)
- Standard technique for real-time sound synthesis

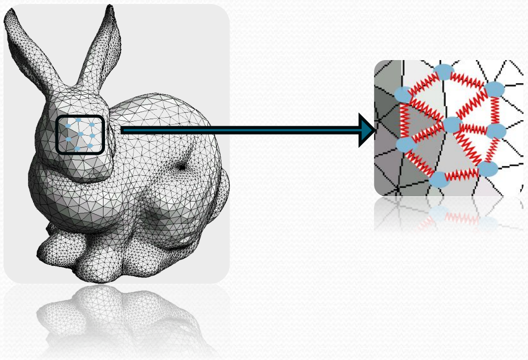
Modal Analysis

- Discretization
  - An input triangle mesh -> a spring-mass system
  - A spring-mass system -> a set of decoupled modes



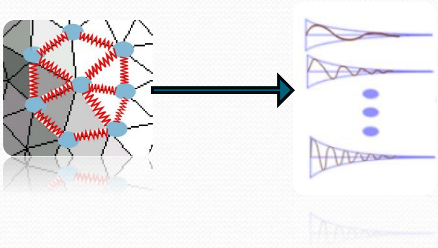
Modal Analysis

- The spring-mass system set-up
  - Each vertex is considered as a mass particle
  - Each edge is considered as a damped spring



Modal Analysis

- Coupled spring-mass system to a set of decoupled modes

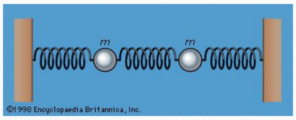


Modal Analysis

- A discretized physics system
  - We use spring-mass system

$$\boxed{\boxed{K}(d)+\boxed{C}(d,\dot{d})+\boxed{M}(\ddot{d})=f}$$

Stiffness Damping Mass



©1998 Encyclopaedia Britannica, Inc.

- Small displacement, so consider it linear

$$\boxed{K}d+\boxed{C}\dot{d}+\boxed{M}\ddot{d}=f$$

Stiffness Damping Mass

Modal Analysis

- Solve the Ordinary Differential Equation (ODE)

$$Kd + C\dot{d} + M\ddot{d} = f$$

- Rayleigh damping

$$Kd + (\gamma M + \eta K)\dot{d} + M\ddot{d} = f$$

And diagonalizing

$$K = GDG^{-1}$$

- Now, solve this ODE instead

$$DG^{-1}d + (\gamma G^{-1}M + \eta DG^{-1})\dot{d} + G^{-1}M\ddot{d} = G^{-1}f$$

Modal Analysis

- Solve the ODE

$$DG^{-1}d + (\gamma G^{-1}M + \eta DG^{-1})\dot{d} + G^{-1}M\ddot{d} = G^{-1}f$$

- Substitute $z = G^{-1}d$ the modes)

Now, solve this ODE instead

$$Dz + (\gamma M + \eta D)\dot{z} + M\ddot{z} = G^{-1}f$$

Modal Analysis

- General solution

$$z_i = c_i e^{\omega_i^+ t} + \bar{c}_i e^{\omega_i^- t}$$

$$\omega_i^{\pm} = \frac{-(\gamma\lambda_i+\eta)\pm\sqrt{(\gamma\lambda_i+\eta)^2-4\lambda_i}}{2}$$

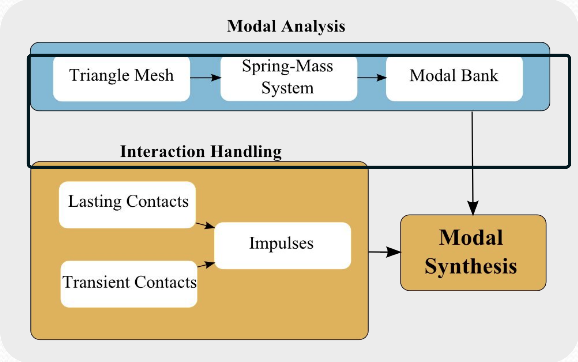
  - External excitation defines the initial conditions

Modal Analysis

- Assumptions
  - In most graphics applications, only surface representations of geometries are given
  - A surface representation is used in modal Analysis
  - Synthesized sound appears to be “hallow”

Modal Analysis Summary

- An input triangle mesh ->
  A spring-mass system ->
  A set of decoupled modes


Modal Analysis

Triangle Mesh → Spring-Mass System → Modal Bank

Interaction Handling

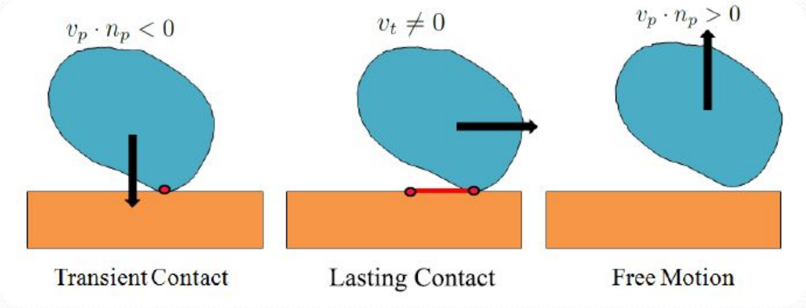
Lasting Contacts

Transient Contacts

Impulses

Modal Synthesis

State Detection



State Detection

- Distinguishing between lasting and transient contacts
  - In contacts?

$$\begin{cases} v_p \cdot n_p < 0 & \text{in contact} \\ v_p \cdot n_p > 0 & \text{not in contact} \end{cases}$$

  - In lasting contacts?

$$\begin{cases} v_t \neq 0 & \text{lasting contact} \\ v_t = 0 & \text{not in lasting contact} \end{cases}$$

Interaction Handling

- Lasting contacts -> a sequence of impulses
- Transient contacts -> a single impulse

Impulse Response

- Dirac Delta function as impulse excitation
  - General solution

$$z_i = c_i e^{\omega_i^+ t} + \bar{c}_i e^{\omega_i^- t}$$

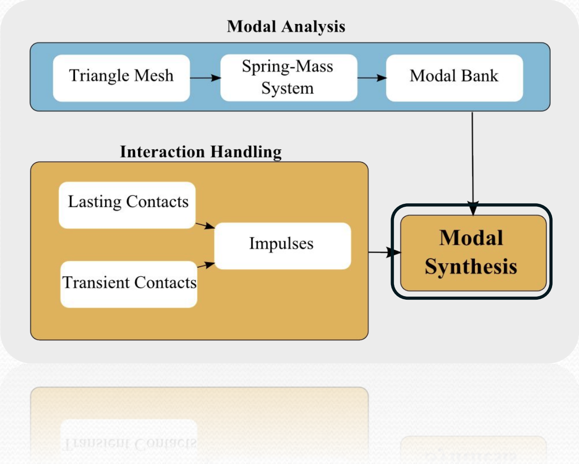
with initial condition given by the impulse, we have:

$$c_i' = c_i e^{\omega_i^+ t_0} + \frac{g_i}{m_i(\omega_i^+ - \omega_i^-)}$$

$$\bar{c}_i' = \bar{c}_i e^{\omega_i^- t_0} - \frac{g_i}{m_i(\omega_i^+ - \omega_i^-)}$$

Impulse Response

$$z_i = c_i e^{\omega_i^+ t} + \bar{c}_i e^{\omega_i^- t}$$



Modal Analysis

Triangle Mesh → Spring-Mass System → Modal Bank

Interaction Handling

Lasting Contacts → Impulses

Transient Contacts → Impulses

Impulses → **Modal Synthesis**

Modal Bank → **Modal Synthesis**

Handling Lasting Contacts

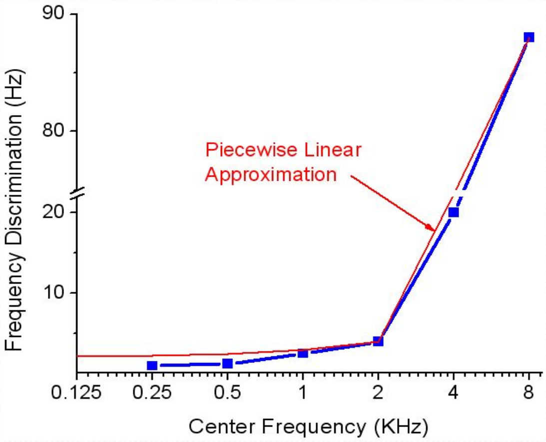
- The interaction simulation has to be stepped at the audio sampling rate: 44100 Hz
- The update rate of a typical real-time physics simulator: on the order of 100's Hz
- Not enough simulation is provided by the physics engine
- An customized interaction model for sound synthesis

Mode Compression: Principle

- Humans can't distinguish two frequencies arbitrarily close to each other [Sek et. al., 1995*]
- Accuracy in discriminating frequencies depends on the frequency in question
- Different frequencies were played in succession to find if the subject could distinguish between them

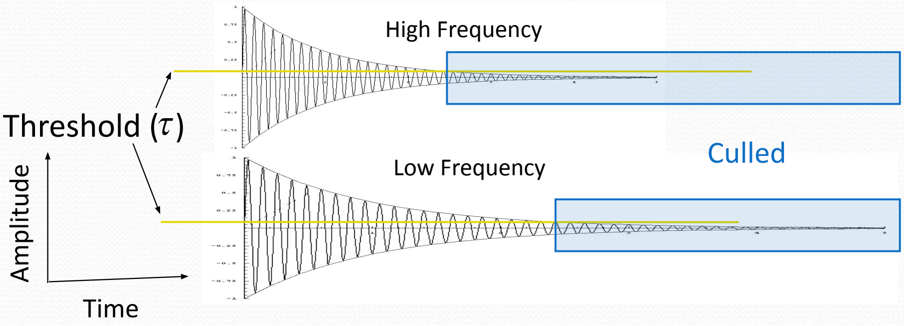
*Sek, A., and Moore, B. C. 1995. Frequency discrimination as a function of frequency, measured in several ways. J. Acoust. Soc. Am. 97, 4 (April), 2479–2486.

Mode Compression: Auditory Perception



Frequency Discrimination at 2 KHz is about 4 Hz -- We can't tell apart frequencies within the range 1998 - 2002 Hz -> *playing many sinusoids of different frequencies together*

Mode Truncation



- Impact Sounds: Attack + Decay
- Key Point: Critical to capture attack properly
- Stop mixing mode when its contribution falls below a prescribed threshold, τ (typically -60 to -80 dB of initial level)

Quality Scaling

- A typical audio scene consists of foreground and background sounds
- *Higher intensity sounds* are considered to be foreground
- Idea: *Give more importance to foreground sounds*
- Provides a graceful way to adapt to variable time constraints

---

System Demonstration

<u>VIDEO</u>

Sound from Image Textures

VIDEO

Limitations

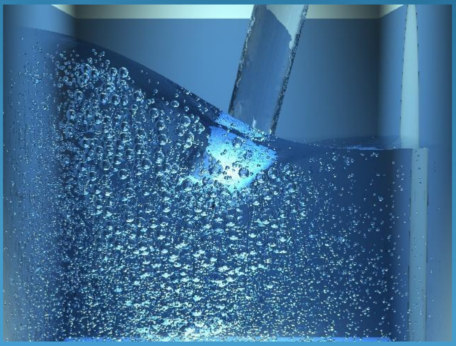
- Implementation: Model only the surface (not an inherent limitation of the approach)
- More approximate than an FEM-based formulation
  - Some tuning is required
- All sound synthesis techniques relying on Modal Analysis: Can only use linear damping models

Sounding Liquids [Moss et al. 2009]

- Work in physics & engineering literature since 1917
  - Sound generated by resonating bubbles
- *Physically-based Models for Liquid Sounds* (van den Doel, 2005)
  - Spherical bubble model
  - No fluid simulator coupling
    - Hand tune bubble profile

Background (Fluid)

- Grid-based methods
- Accurate to grid resolution
  - Bubbles can be smaller
  - Slow
  - Can be two-phase



Background (Fluid)

- Shallow Water Equations
  - Simulate water surface
    - No breaking waves
  - Real time
  - One phase
    - Explicit bubbles

Overview

- Generate sound from existing fluid simulation
  - Model sound generated by bubbles
- Apply model to two types of fluid simulators

- **Particle-Grid-based**
  - Extract bubbles
  - Process spherical and non-spherical bubbles
  - Generate sound

- **Shallow Water Equations**
  - Processes surface
    - Curvature and velocity
  - Select bubble from distribution
  - Generate sound

Mathematical Formulations

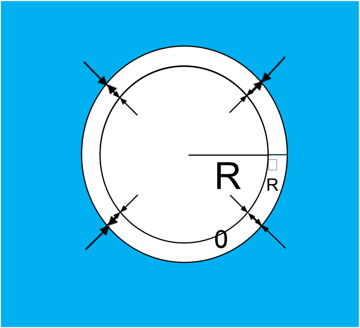
- **Spherical Bubbles**

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{3\gamma p_0}{\rho R_0^2}}$$

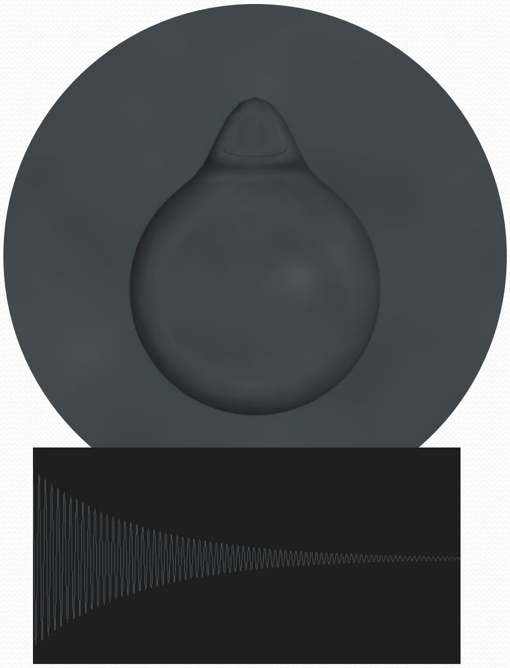
$$\tau(t) = Asin(2\pi f(t)t)e^{-dt}$$

- **Non-spherical bubbles**
  - Decompose into a spherical harmonics

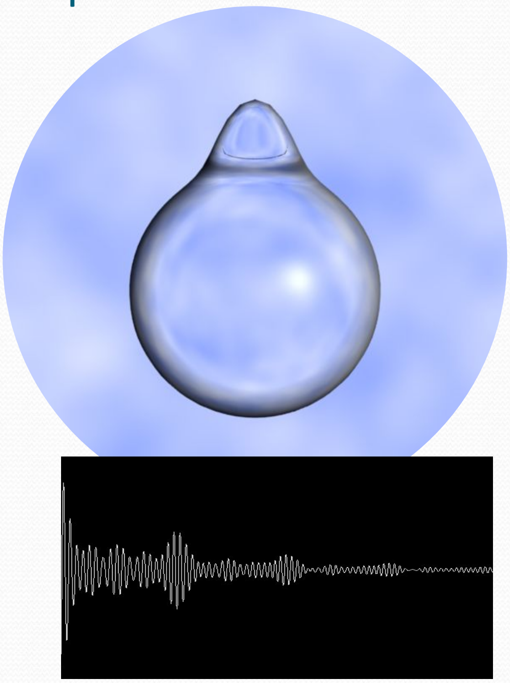
$$f_n^2 \approx \frac{1}{4\pi^2}(n-1)(n+1)(n+2)\frac{\sigma}{\rho R_0^3}$$



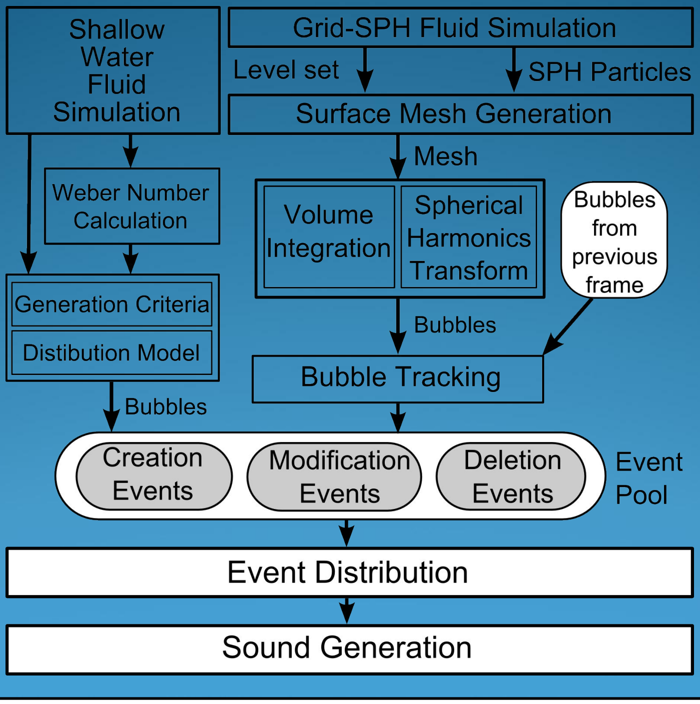
Without Spherical Harmonics



With Spherical Harmonics



System Overview



Shallow Water Fluid Simulation

Grid-SPH Fluid Simulation

Level set

SPH Particles

Surface Mesh Generation

Mesh

Weber Number Calculation

Volume Integration

Spherical Harmonics Transform

Bubbles from previous frame

Generation Criteria

Distibution Model

Bubbles

Bubble Tracking

Bubbles

Creation Events

Modification Events

Deletion Events

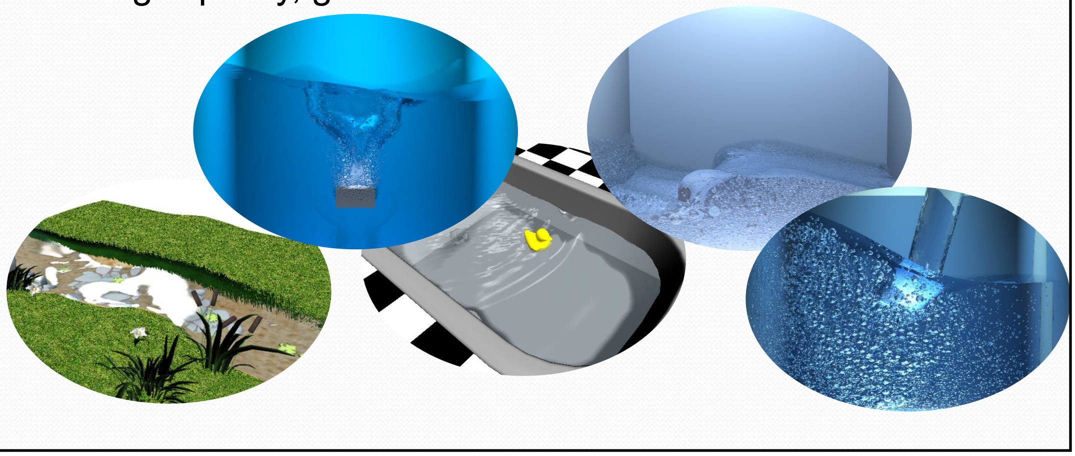
Event Pool

Event Distribution

Sound Generation

Summary

- Simple, automatic sound synthesis
- Applied to two fluid simulators
  - Interactive, shallow water
  - High-quality, grid based



Video Demonstration

VIDEO

Overview

- Interactive Sound Synthesis
- Efficient Numerical Acoustic Simulation
- Interactive Sound Propagation
- Conclusion and Future Work

Overview

- Interactive Sound Synthesis
- Efficient Numerical Acoustic Simulation
  - Novel technique based on 3D Adaptive Rectangular Decomposition
  - Hundred times faster than Finite Difference Time Domain
- Interactive Sound Propagation
- Conclusion and Future Work

Acoustics: Governing Equation

- Solve the Linear Wave Equation:

$$\frac{\partial^2 p}{\partial t^2} - c^2 \nabla^2 p = F(\mathbf{x}, t)$$

∇^2 is the Laplacian operator in 3D

c is the speed of sound in air

p is the pressure field to solve

- The RHS is the forcing term, corresponding to volume sound sources in the scene

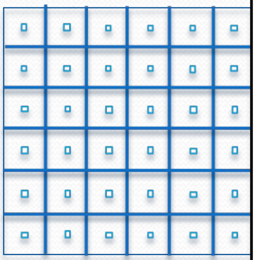
State of the art: Room Acoustics

- Geometric Techniques (e.g. Ray/Beam/Frustum Tracing) combined with explicit diffraction modeling
- Auralization software (e.g. ODEON, CATT): Hybrid image-source and ray-tracing along with (upcoming) explicit diffraction modeling
- Numerical acoustic simulation for complex 3D scenes has been explored only since mid-2000 (~2006)

Acoustics for Interactive Applications

- Geometric Approaches
    - Beam Tracing (Funkhouser et. al.)
    - Phonon Tracing (Bertram and Deines et. al.)
    - Frustum Tracing (Chandak et. al.)
-
- Advantages: Efficient, easy to understand
-
- Difficulties: Diffraction / Scattering, high-order reflections

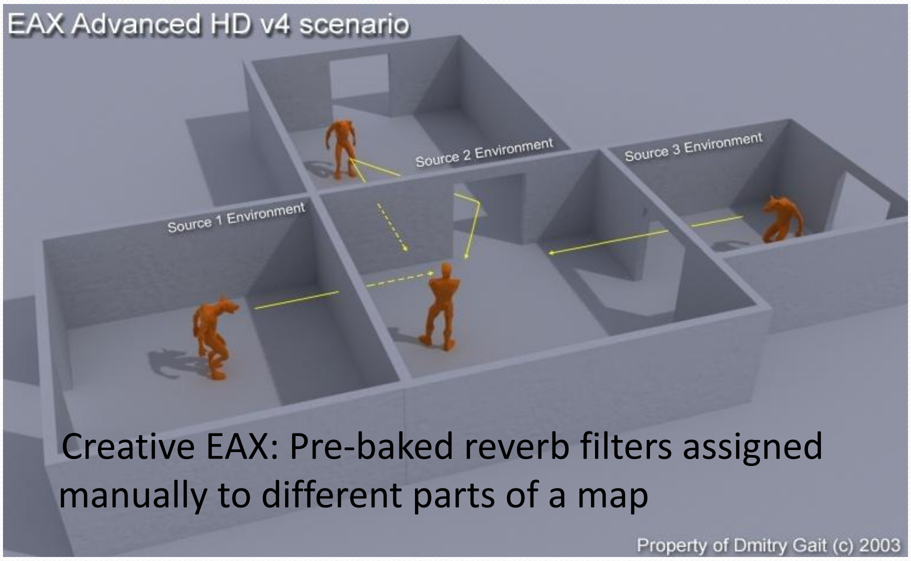
Numerical Acoustics



- Discretize and solve Wave Equation on a grid
- Explored for complex 3D scenes (eg. auditoria) only recently (2004 – 2006) by Sakamoto et. al.
- Disadvantage: Slow and memory-intensive
  - Simulations are band-limited
  - ○
- Advantages: Diffraction / Scattering, high-order reflections

---

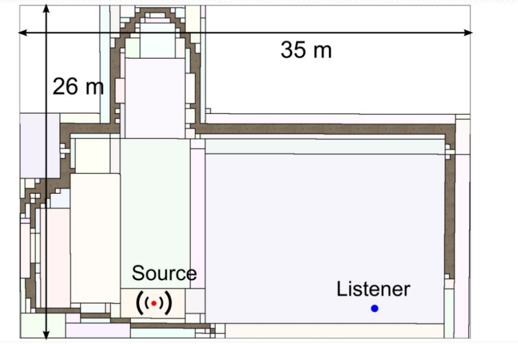
Acoustics in Games

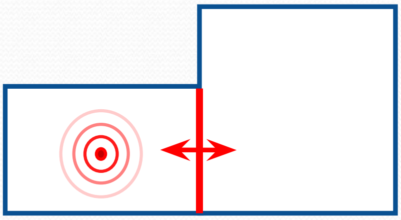


Creative EAX: Pre-baked reverb filters assigned manually to different parts of a map

Property of Dmitry Gait (c) 2003

Adaptive Rectangular Decomposition





Interface Handling

- Numerical Simulation of the Wave Equation
- Rectangular Decomposition of a 3D scene
- Exploit analytical solutions on rectangular spaces
- 6th order Finite Difference for interface transmission

Solution on a Rectangular Domain

- Rectangular space in 3D with size (l_x, l_y, l_z), and perfectly reflective boundary
- Modal Analysis can be done ***analytically*** –

$$p(x,y,z,t) = \sum_{i=(i_x,i_y,i_z)} m_i(t)\,\Phi_i(x,y,z)$$

$$\Phi_i(x,y,z) = \cos\left(\frac{\pi i_x}{l_x}x\right)\cos\left(\frac{\pi i_y}{l_y}y\right)\cos\left(\frac{\pi i_z}{l_z}z\right)$$

Leveraging GPU for Acoustics

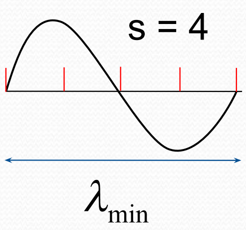
- Solution of Wave Equation within each rectangle can be done using a 3D Discrete Cosine Transform (DCT)
- DCTs can be computed using FFT
- Use efficient FFT implementation on GPU
  - Govindaraju, N. K., Lloyd, B., Dotsenko, Y., Smith, B., and Manferdelli, J. **High-Performance Discrete Fourier Transforms on Graphics Processors**. In the *Proc. of 2008 ACM/IEEE Supercomputing*

Computational Efficiency

- For a scene of size **L** in 3D and simulation duration **T** –

Memory : $\left(\frac{L\underline{s}v_{\max}}{c}\right)^{\boxed{3}}$

Time : $T\left(\frac{L}{c}\right)^{3}\left(\underline{s}v_{\max}\right)^{\boxed{4}}$



- Nyquist Limit: **s ≥ 2**
- FDTD: **s = 10**. My approach: **s = 2.6**
- Speedup with my technique : $(10/2.6)^4$ > 100

Demo

Video

Performance Comparison

Scene Name	Volume (m^3)	Time: FDTD (CPU)	Time: My Technique (GPU)	**Speedup**
Corridor	375	365 min	4 min	**~ 90x**
House*	1,275	3.5 days	24 min	**~ 200x**
Cathedral	13,650	1 week (estimated)	29 min	**~ 300 x**

- Quad-core 2.8GHz Intel Xeon CPU with 8GB RAM, NVIDIA GeForce GTX 280

* This simulation was band-limited to 2 kHz, instead of 1 kHz

Summary

- Adaptive Rectangular Decomposition yields 100x improvement in performance over FDTD and consumes 10x less memory
- Source of Speedup: Modal Analysis of rectangular spaces as well as GPU-DCT
- Can feasibly simulate acoustics for large, complex scenes, such as a Cathedral

---

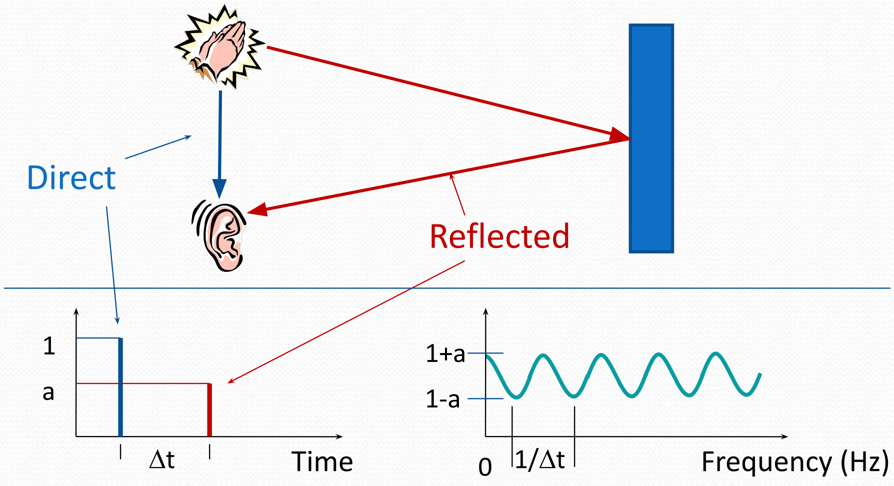
Overview

- Interactive Sound Synthesis
- Efficient Numerical Acoustic Simulation
- Interactive Sound Propagation
- Conclusion and Future Work

Overview

- Interactive Sound Synthesis
- Efficient Numerical Acoustic Simulation
- Interactive Sound Propagation
  - Perceptual aspects of acoustics
  - Novel perceptually-motivated techniques
  - Interactive auralization system: moving sources and listener
- Conclusion and Future Work

Impulse Response (IR)

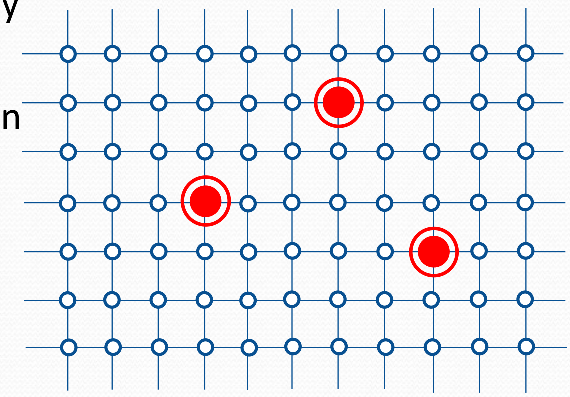


Impulse Response

Frequency Response

Challenges

- Direct approach is costly
- Days of simulation (even with fast simulator)
- Terabytes of storage



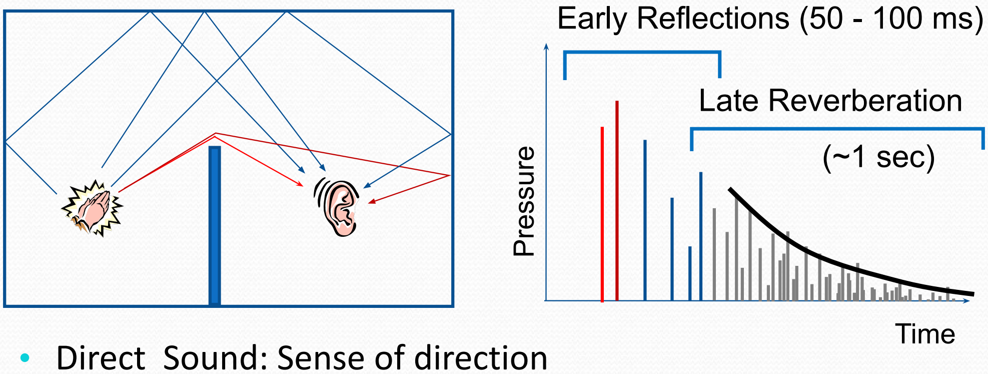
● Source locations

○ Listener locations

Contributions

- Approach –
  - Sample data at lower resolution in space (~1 m)
  - Novel perceptually-based scheme: Store each Impulse Response compactly
  - Spatially-interpolate Impulse Responses
- Audio engine that uses fast frequency-domain convolutions

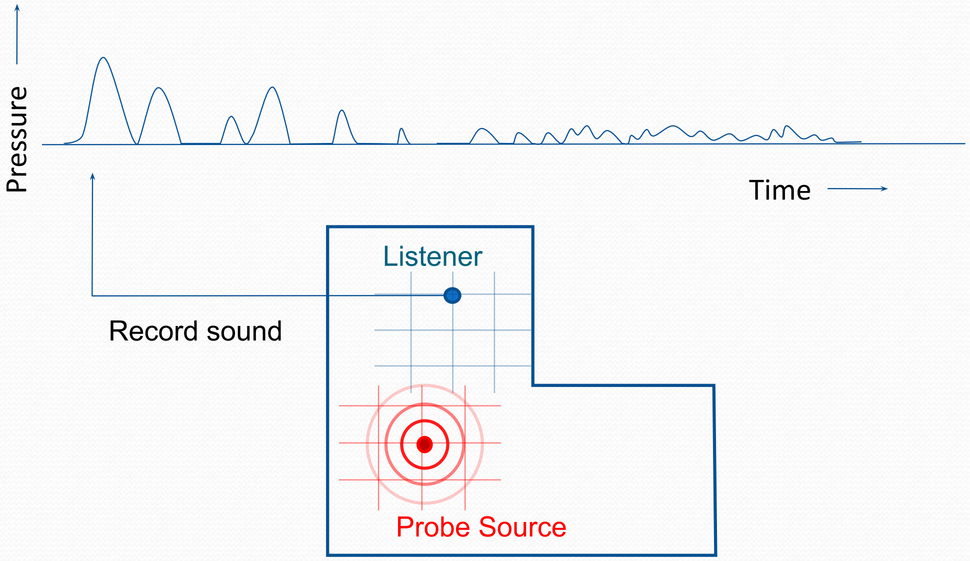
Auditory perception of acoustic spaces



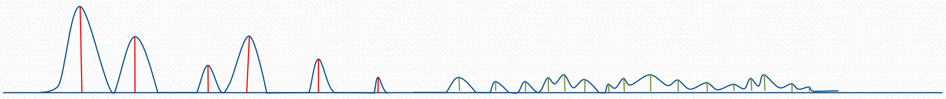
- Direct Sound: Sense of direction
- ER: Loudness, Timbre, “Envelopment”. Perceivable spatial variation.
- LR: Only statistical properties perceivable – Decay Time (RT60), Periodicities (Flutter echoes)

Reference: “Room Acoustics” by Heinrich Kuttruff

IR Factoring (1)



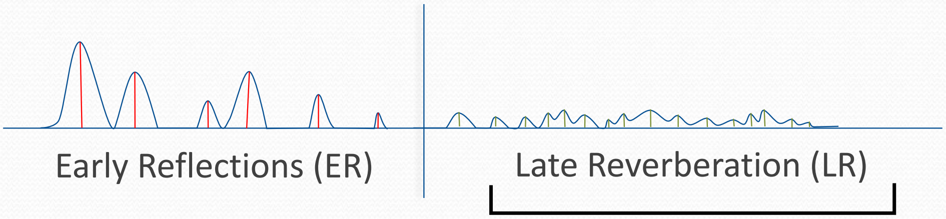
IR Factoring (2)



Peak Detection

- Finds peak delays and amplitudes

IR Factoring (3)



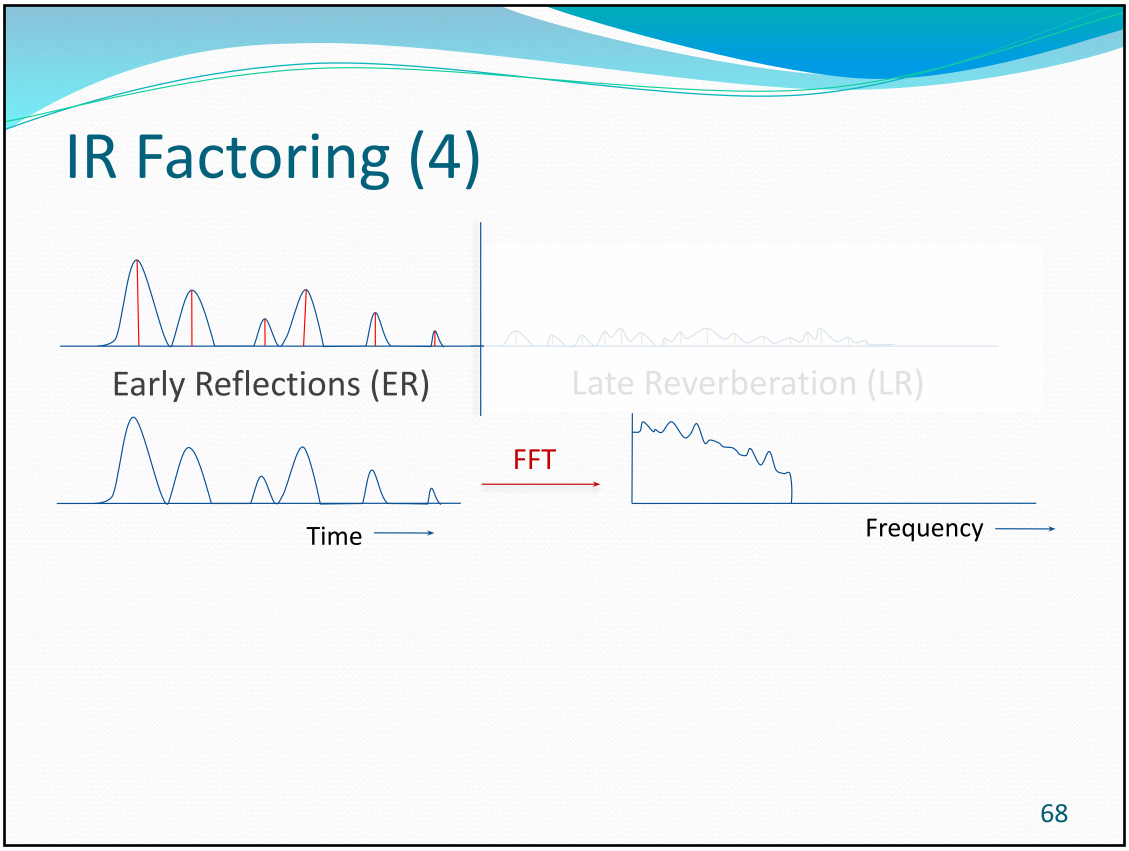
Early Reflections (ER)

Late Reverberation (LR)

Store peaks as LRIR

ER - LR Decomposition

- Compute and store only one Late Reverberation filter per room
- Reduces pre-computation time and runtime memory usage by 10 times



IR Factoring (4)

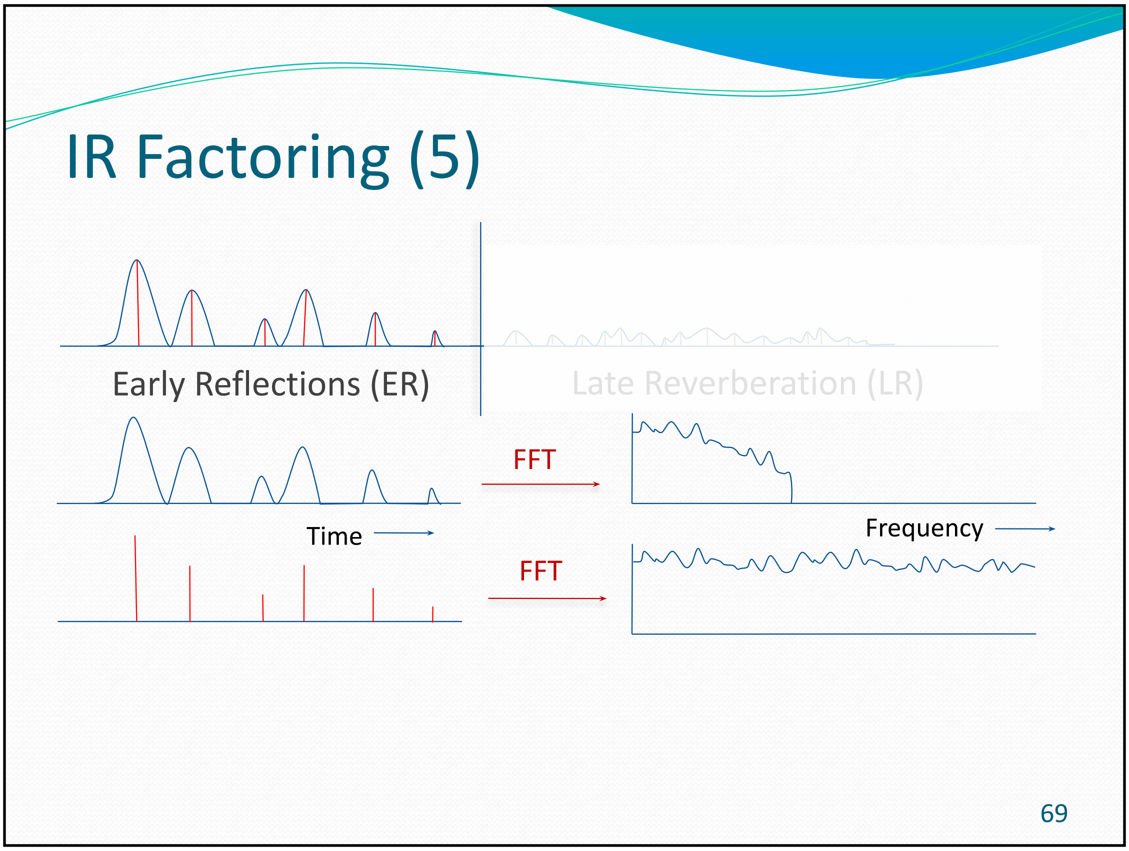
Early Reflections (ER)

Late Reverberation (LR)

FFT

Time

Frequency



IR Factoring (5)

Early Reflections (ER)

Late Reverberation (LR)

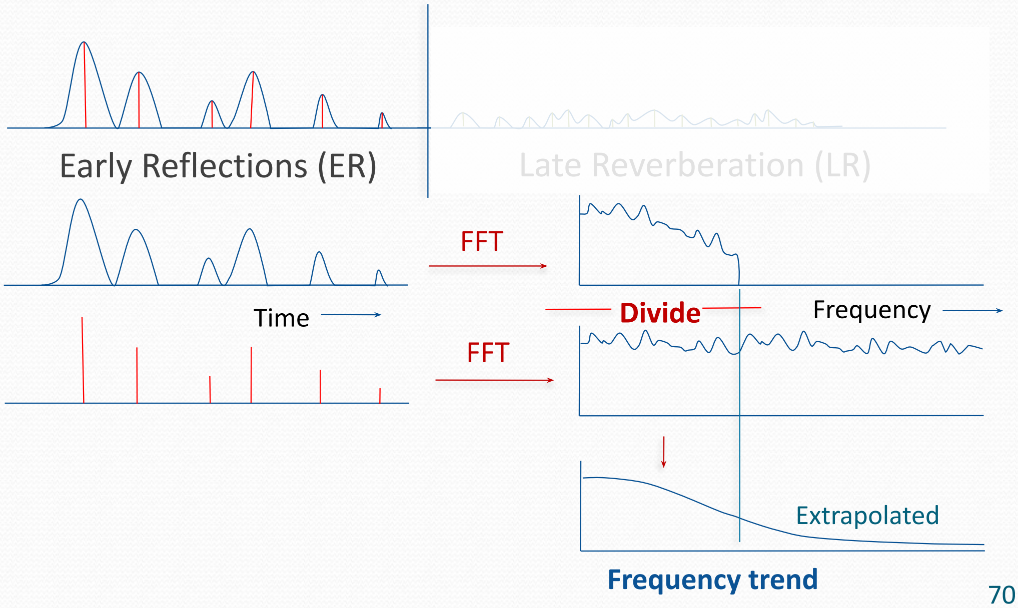
FFT

Time

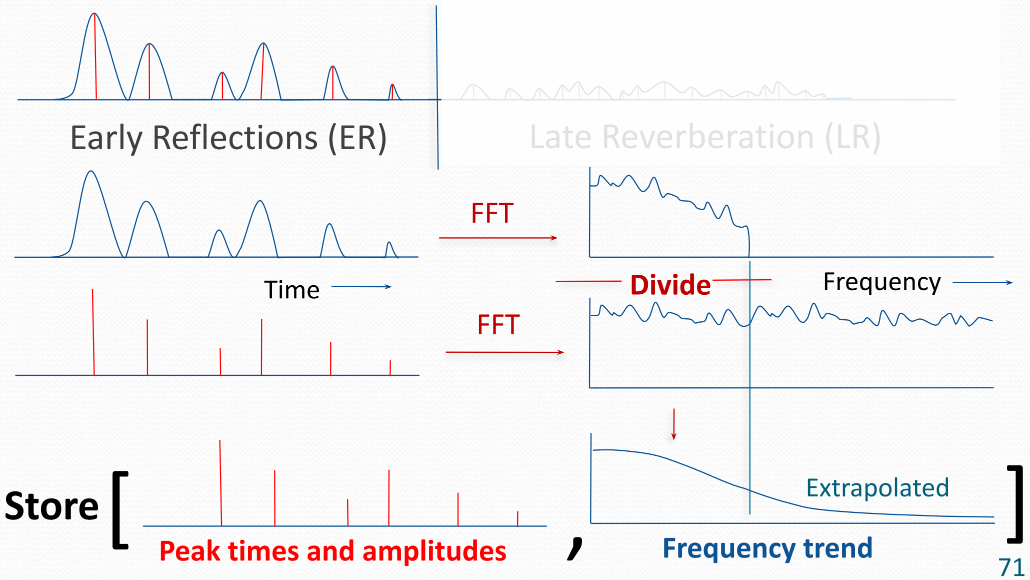
Frequency

FFT

IR Factoring (6)

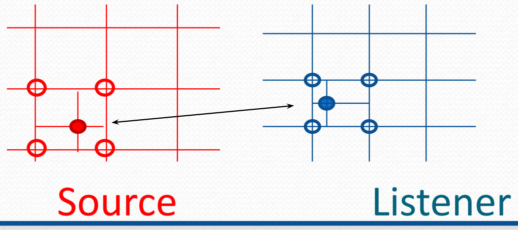


IR Factoring (7)



Runtime Processing

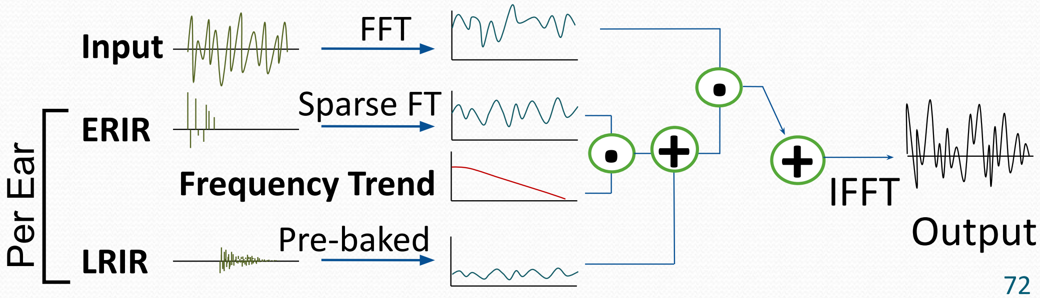
ERIR Interpolation



Source

Listener

Auralization



Input

FFT

Per Ear

ERIR

Sparse FT

Frequency Trend

LRIR

Pre-baked

IFFT

Output

Performance

- Pre-processing times typically a few hours
- Can handle about 10 sound sources in real-time on a Quad-core Xeon system with 4 GB RAM
- Bottleneck: 1D FFT
  - Auralization system maps well to parallel processors

Walkthrough: Game Scene

- “Citadel” Scene from the game Half-Life 2
- Large Size: 3,500 m^3
- Complex geometry (fin-like structures)

System Demonstration

Video

http://www.youtube.com/watch?v=MQt1jtDBNK4

Summary

- The first interactive sound propagation system that leverages numerical simulation
- Can render important acoustic effects like Late Reverberation and Diffraction low-pass filtering in real-time
- Can handle multiple moving sound sources and listener
- Works for large, complex 3D scenes

Overview

- Sound Synthesis
- Efficient Numerical Acoustic Simulation
- Interactive Sound Propagation
- Conclusion and Future Work

Summary

- Interactive Sound Synthesis
  - Perceptually-motivated optimizations enabling hundreds of sounding objects in real-time
- Efficient Numerical Acoustics
  - A simulator 100 times faster than Finite Difference Time Domain for constant wave speed simulations
- Interactive Sound Propagation
  - Leverage fast numerical acoustics
  - Exploit auditory perception
  - Render wave-based acoustics for multiple moving sources and listener in real-time

Conclusion

- Physically-based Sound: Complex underlying physical processes require a lot of computational power
- Combination of efficient algorithms, perceptually-motivated optimizations and fast hardware

Recent Work: Sound Synthesis

- **Ultimate goal**: Virtual Worlds with physically-based sounds for collisions, rolling, sliding, creaking, cloth, gunshots, water, automobiles, and so on
- Infer audio materials from video
- Virtual Musical Instruments using next-gen UI
- Mobile Musical Instruments

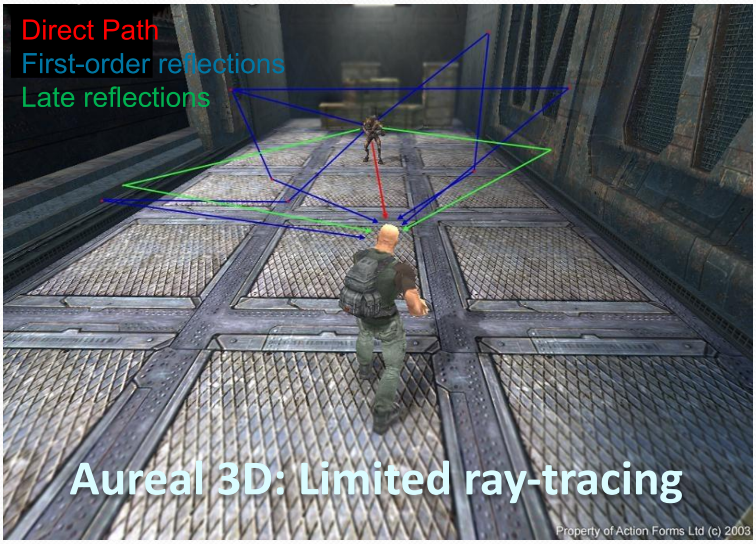
Recent Work: Sound Propagation

- Acoustics for Games and Virtual Worlds
- Accurate numerical predictions in auditorium design
- Efficient numerical solvers for high-performance computing applications
- Combine Sound Synthesis and Acoustics for a completely physically-based auralization system

References

- Nikunj Raghuvanshi and Ming C. Lin. **Interactive Sound Synthesis for Large Scale Environments**. In SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games, 2006.
- Nikunj Raghuvanshi, Nico Galoppo, and Ming C. Lin. **Accelerated Wave-based Acoustics Simulation**. In SPM ’08: Proceedings of the 2008 ACM Symposium on Solid and physical modeling, 2008.
- Nikunj Raghuvanshi, Rahul Narain, and Ming C. Lin. **Efficient and Accurate Sound Propagation using Adaptive Rectangular Decomposition**. IEEE Transactions on Visualization and Computer Graphics, December 2009.
- Nikunj Raghuvanshi, Brandon Lloyd, and Ming C. Lin. **Efficient Numerical Acoustic Simulation on Graphics Processors Using Adaptive Rectangular Decomposition**. Proc. EAA Symposium on Auralization, 2009.

Acoustics in Games

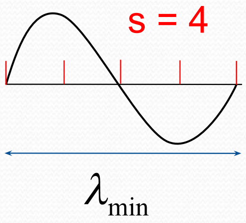


Property of Action Forms Ltd (c) 2003

Computational Challenges (1)

- Need sufficient spatial resolution to resolve smallest wavelength of interest

$$h \sim \frac{\lambda_{\min}}{s} \quad , \ s > 2$$



- Also, need sufficiently small time-step to resolve highest frequency

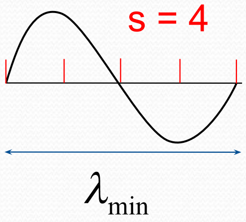
$$dt < \frac{1}{s\nu_{\max}}$$

Computational Challenges (2)

- For a scene of size **L** in 3D and simulation duration **T** –

Memory : $\left(\frac{Ls\nu_{\max}}{c}\right)^3$

Time : $T\left(\frac{L}{c}\right)^3\left(s\nu_{\max}\right)^4$

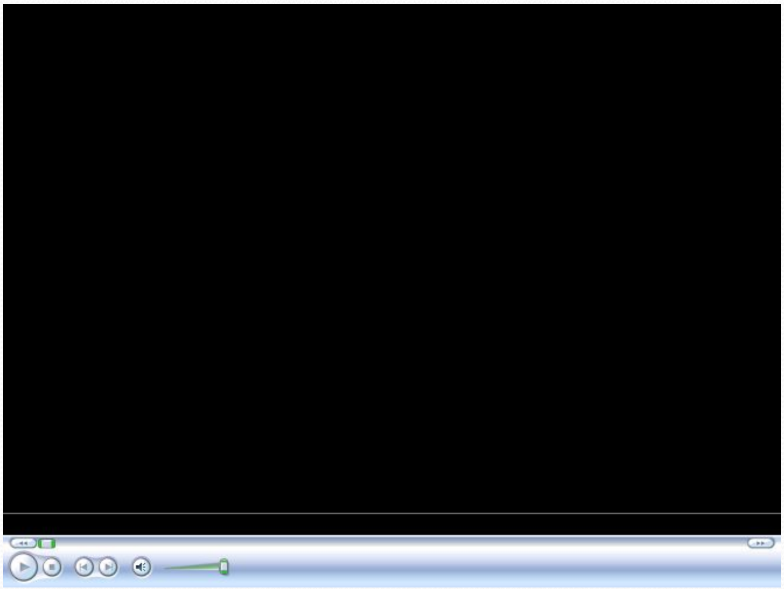


- For a medium-sized room – L = 10 m, T = 1 s, c = 340 m/s, s = 10, $\nu_{\max}$ = 10,000 Hz

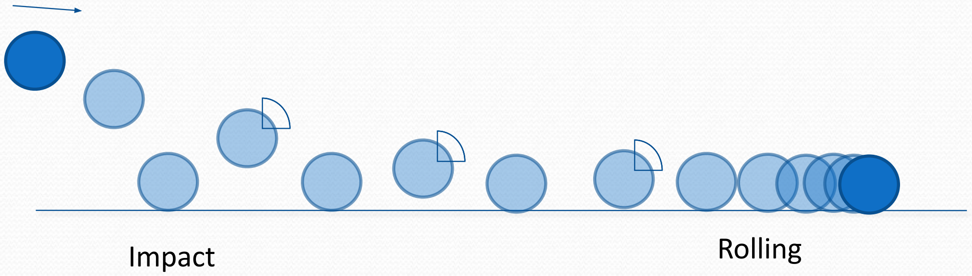
Memory : **~100 GB** Time: **6 days**, at 100 GFLOPS

Bass Boost in small spaces

Input



A typical scenario

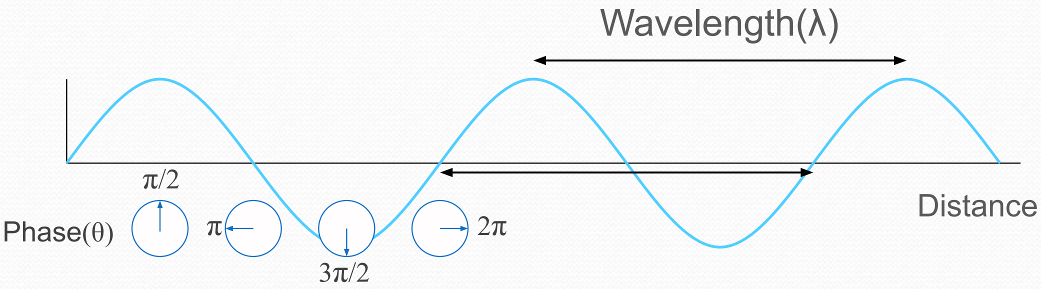


- How do you handle this with recorded sounds?

Related/Future Work

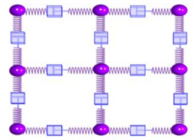
- Model-based Synthesis
  - Current work with Naga K. Govindaraju, Brandon Lloyd, Guy Whitmore and Chirstopher Melroth
- Sliding Sounds
  - Past work in SIGGRAPH by Doel et. al.
  - Current work being done by Zhimin Ren at UNC
- Liquid Sounds
  - Recent paper on “Harmonic Fluids” by Doug James at Cornell
  - Also, some work at UNC by Yero Yeh
- Cloth Rustling

Quick primer on waves



- Phase (θ): Measures the progression of wave between crest and trough
- Frequency: ν , Wavelength: λ
- Wave Speed , $c = \nu\lambda$

Equation of Motion



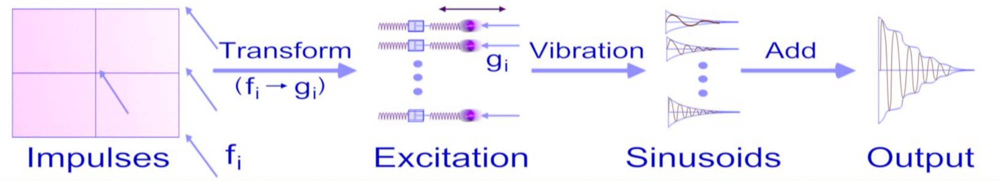
Spring-Mass System

- Equation of motion (linear system of coupled ODEs):

$$\underbrace{M\frac{d^2r}{dt^2}}_{\text{Inertia}} + \underbrace{(\gamma M + \eta K)\frac{dr}{dt}}_{\text{Damping}} + \underbrace{Kr}_{\text{Elasticity}} = \underbrace{F(t)}_{\text{Force}}$$

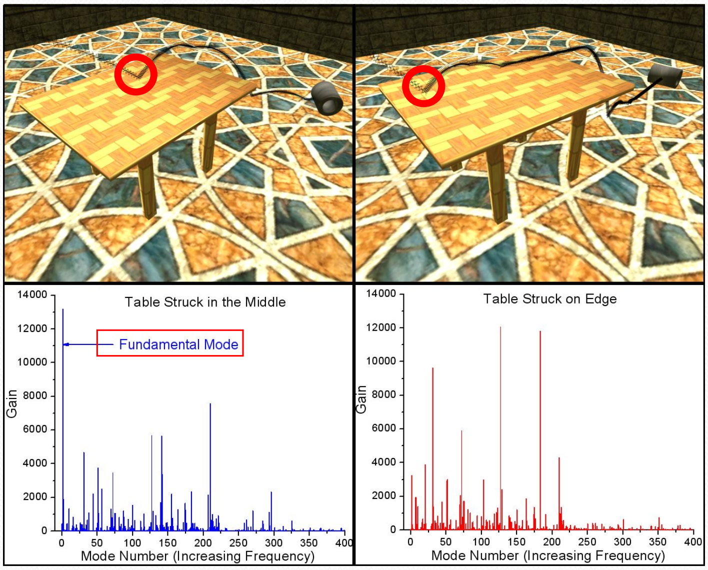
γ , η : Fluid and Viscoelastic Damping constants

Sound Synthesis

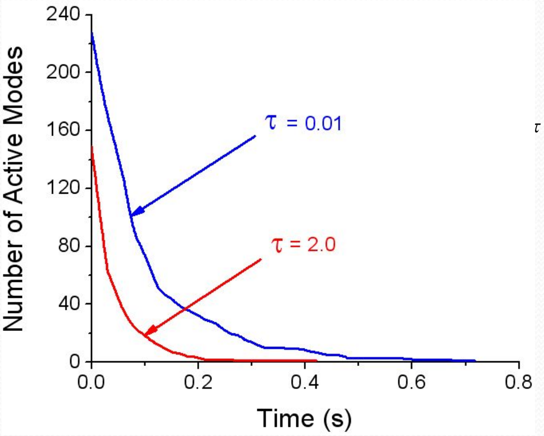

Transform
$(f_i \rightarrow g_i)$
g_i
Vibration
Add
Impulses
f_i
Excitation
Sinusoids
Output

- Rigid Body Simulator provides impulses
- Transform to mode gains
- Sound synthesized by adding the modes' sinusoids
- Advantage: Adding damped sinusoids is very fast

Position Dependent Sounds: Analysis

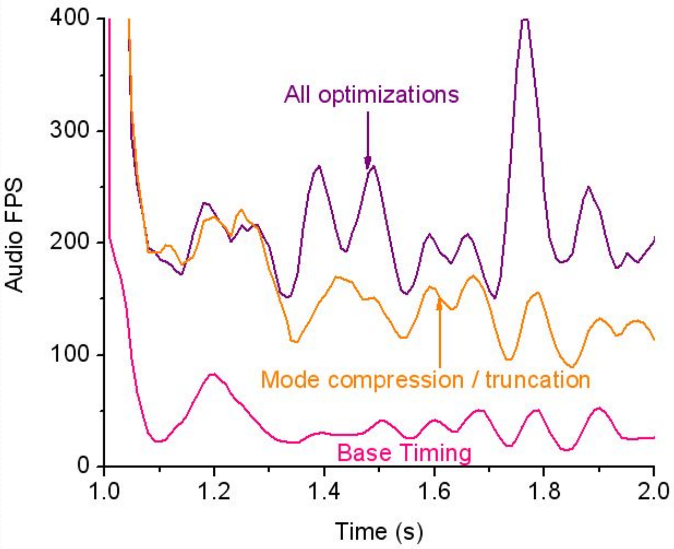


Mode Truncation: Performance



- Xylophone bar struck in the middle
- Higher value of τ: Modes fall off more quickly
- Very little perceptual difference

Efficiency: Analysis

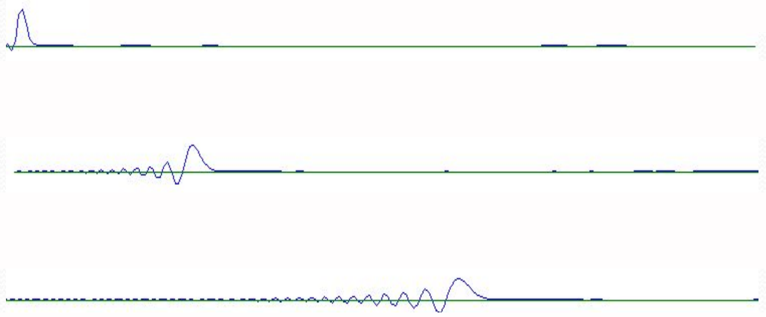


Acoustics: Computational Challenges

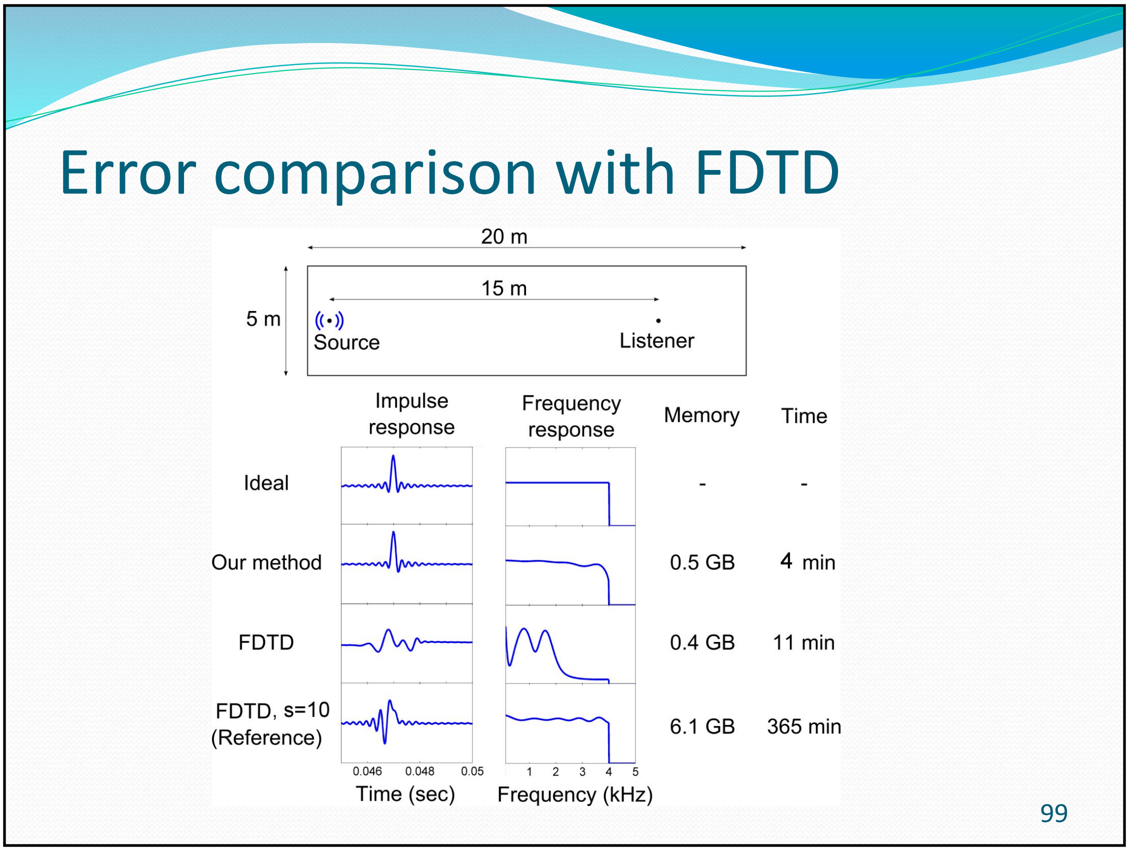
- Multiple reflections are audible: Full time domain solution required, unlike lighting
- Interference is important eg. Dead spots in auditoria
- Diffraction is observable for sound and must be captured properly

---

Errors in FDTD: Numerical Dispersion

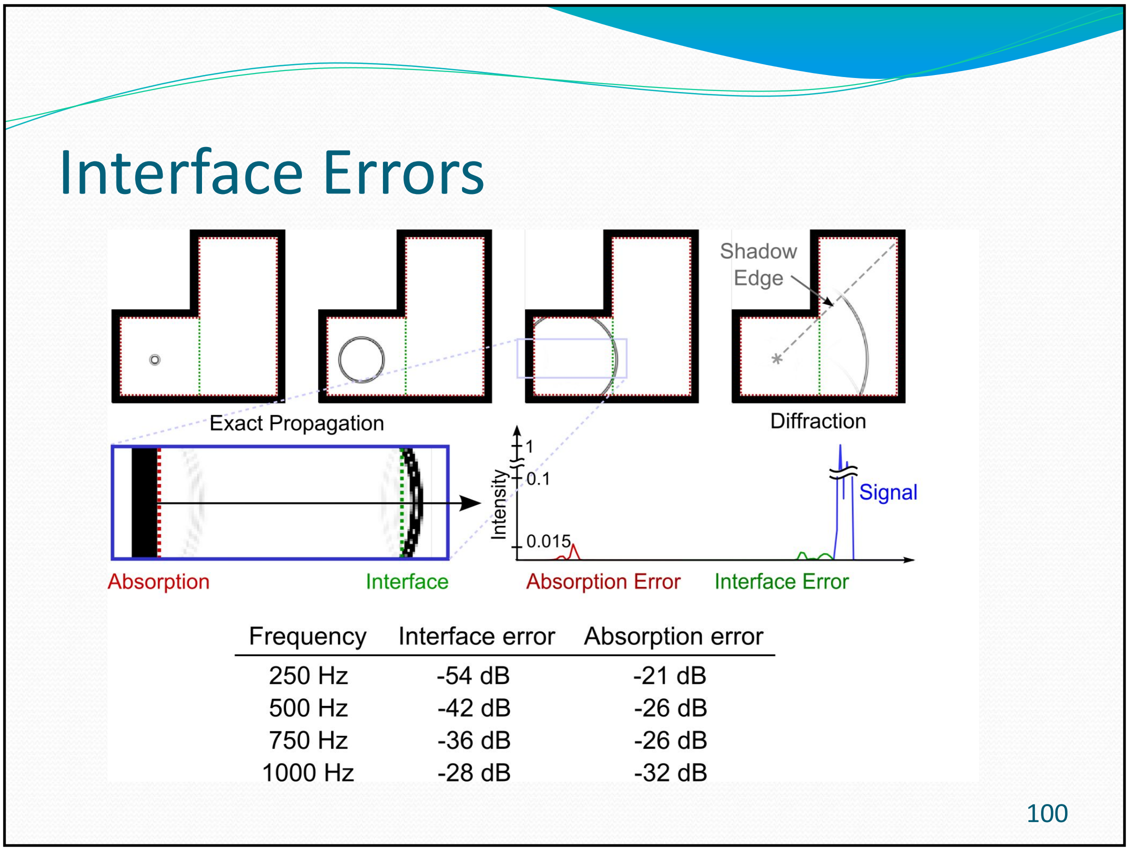


- All frequencies don’t travel with the same numerical speed
- Need s = 10 for this



Error comparison with FDTD

	Memory	Time
Ideal	-	-
Our method	0.5 GB	4 min
FDTD	0.4 GB	11 min
FDTD, s=10 (Reference)	6.1 GB	365 min



Interface Errors

Frequency	Interface error	Absorption error
250 Hz	-54 dB	-21 dB
500 Hz	-42 dB	-26 dB
750 Hz	-36 dB	-26 dB
1000 Hz	-28 dB	-32 dB

Solution on a Rectangular Domain

- Rectangle in 3D with size : (l_x, l_y, l_z), with sound-hard boundary
- Represent pressure in Modal basis for Laplacian on the rectangular domain:

$$p(x, y, z, t) = \sum_{i=(i_x, i_y, i_z)} m_i(t)\, \Phi_i(x, y, z)$$

$$\Phi_i(x, y, z) = \cos\left(\frac{\pi i_x}{l_x} x\right) \cos\left(\frac{\pi i_y}{l_y} y\right) \cos\left(\frac{\pi i_z}{l_z} z\right)$$

$m_i(t)$ are time-varying mode coefficients

Φ_i are eigen-functions of Laplacian

Solution in Spectral Basis using DCT

- The transformation from real space and spectral space can be done using 3D DCT and inverse DCT

$$\{m_i\} = DCT\{p_i\}, \quad \{p_i\} = iDCT\{m_i\}$$

- Wave Equation in spectral space (decoupled):

$$\frac{\partial^2 m_i}{\partial t^2} + c^2 k_i^2 m_i = iDCT(F(t)),$$

- Analytical solution in time:

$$k_i^2 = \pi^2\left(\frac{i_x^2}{l_x^2} + \frac{i_y^2}{l_y^2} + \frac{i_z^2}{l_z^2}\right)$$

$$M_i^{n+1} = 2M_i^n \cos(\omega_i \Delta t) - M_i^{n-1} + \frac{2\widetilde{F^n}}{\omega_i^2}\left(1 - \cos(\omega_i \Delta t)\right)$$

Modeling partially absorbing surfaces

- Numerical absorbers for Wave Equation is a tough challenge
- Perfectly Matched Layer (**PML**) was developed in the Electromagnetic simulation community
- We adapt a time-domain formulation described in
  - Y. S. Rickard, N. K. Georgieva, and W.-P. Huang, "Application and optimization of pml abc for the 3-d wave equation in the time domain," *Antennas and Propagation, IEEE Transactions on*, vol. 51, no. 2, pp. 286-295, 2003

Summary (contd..)

- Automatically handles interference and diffraction
- Parallelizable at multiple granularities: Source positions, Partitions, DCT
- Axis-aligned simulation grid, easy to obtain using voxelization

Current Progress

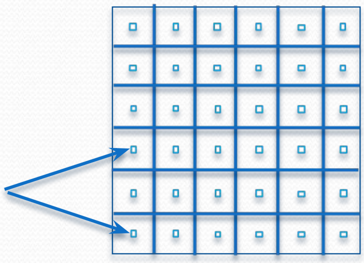
- Compression scheme is nearly finished, can handle a medium-sized Lecture hall
- Extracting diffraction information still needs to be tested and tweaked
- Late reverb interpolation needs to be implemented
- Real-time auralization system is also near completion. IR interpolation needs to be tested properly.

Finite Difference Time Domain (FDTD)

- Discretize continuum derivative operators:

$$\frac{P_{t+dt} - 2P_t + P_{t-dt}}{dt^2} + KP_t + O\left(h^6\right) + O\left(dt^2\right) = F_t$$

- Spatial cell size: h, time-step: dt
- Works on a uniform Cartesian grid
- Pressure sampled at cell centers



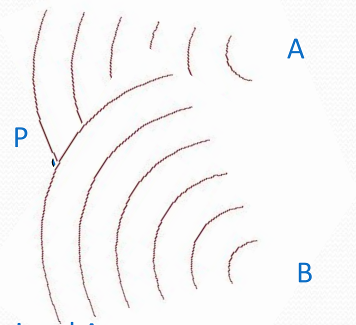
Summary

- Simple formulation and easy to implement
- Works on arbitrary surface meshes
- Acceleration techniques exploiting auditory perception
  - Mode Compression
  - Mode Truncation
  - Quality Scaling
- Well suited for large-scale, real-time applications with stringent time constraints, like games

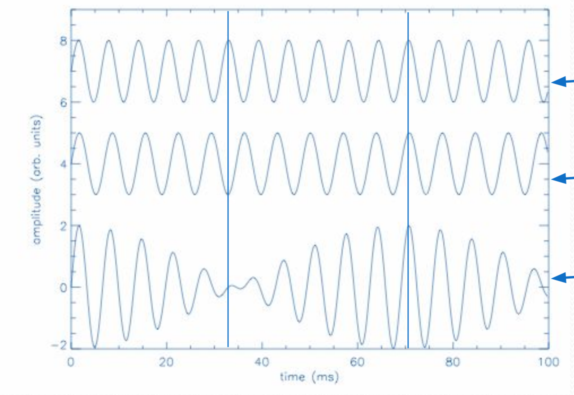
---

Interference

- The resultant pressure at P due to two waves is their sum
- Phase is crucial



out of phase - cancel   in phase - add

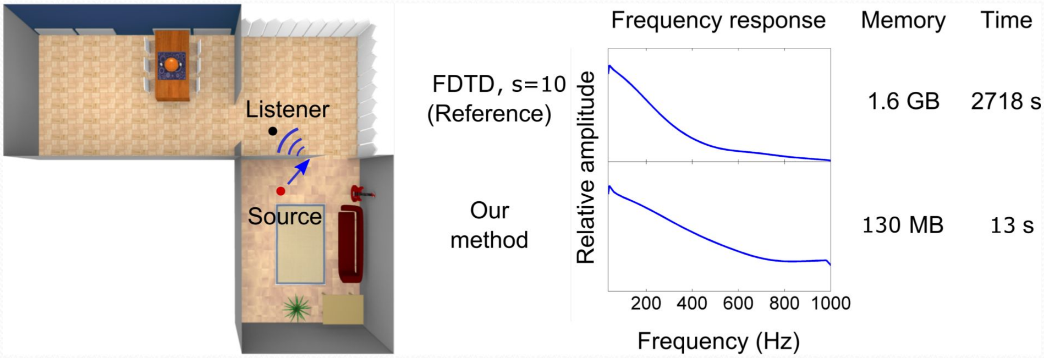


signal A

signal B

A + B

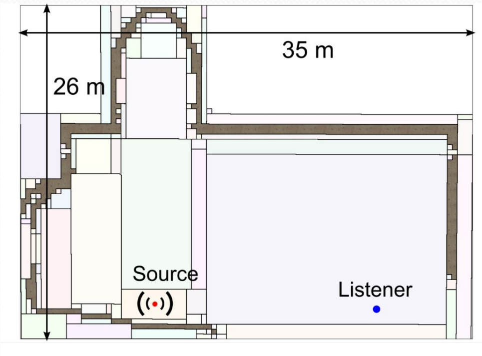
Performance: House

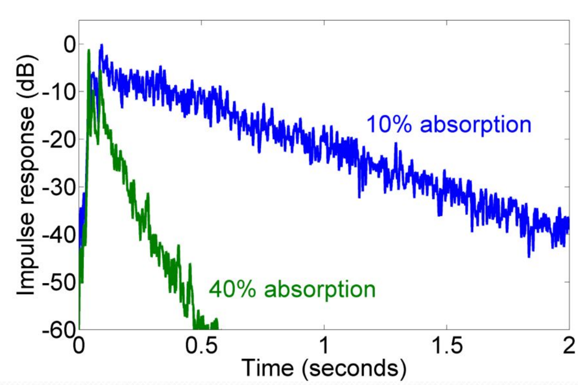


	Memory	Time
FDTD, s=10 (Reference)	1.6 GB	2718 s
Our method	130 MB	13 s

- Dimensions: 17m x 15m x 5 m
- Auralization: 24 minutes to generate a .4 second long Impulse Response (< **2 kHz**)

Performance: Cathedral





- Dimensions: 35m x 15m x 26 m
- 29 minutes to generate a 1-second long Impulse Response (< 1 kHz)