RESound

Interactive Sound Rendering in Dynamic Virtual Environments

Micah Taylor, Anish Chandak
Lakulish Antani, Dinesh Manocha

University of North Carolina
• Sound rendering and applications

• Details of propagation

• Our system: RESound
- Sound rendering and applications
- Details of propagation
- Our system: RESound
Sound Rendering

- Three main steps
  - Signal input
  - Sound propagation
  - Audio output
Sound Rendering: Signal Input

- Recorded sample
  - Simple and fast
  - Played with events
  - Static

- Synthesized sound
  - Physics simulation generates sound
  - Matches virtual events

[Matt Hileo]

[Raghuvanshi 2006]
Sound Rendering: Signal Input

- Synthesized sound
- Uses physical models [Florens et al. 1991]
- Interactive rates with many objects [Raghuvanshi et al. 2006]
- Correlates closely with visual scene [Ren et al. 2009]
Sound Rendering: Propagation

- **Goal:** Model environment influences
  - Echoes
  - Delay from distance
  - Attenuation from distance
  - Frequency shifts
- **Output:** Impulse response
  - Represents room's effect on input signal
Sound Rendering: Propagation

- Common methods
  - No propagation - direct path only
  - Geometric simulation
  - Numerical simulation
Sound Rendering: Audio Output

- **Goal**
  - Combine many sounds from environment
  - Apply any needed effects
  - Output to user's audio device

- **Uses the output from prior steps**
  - Input signal
  - Room impulse response
Sound Rendering: Audio Output

- Common output methods
  - Mono
    - Fast, simple
    - No spatialization
  - Stereo
    - Fast, simple, left+right spatialization
  - 3d sound
    - Head Related Transfer Functions (HRTF)
    - Complex, very good spatialization
Applications

- Video games
  - Helps player avoid monsters
  - Provides sound cues to environment size
  - Used in most 3d video games
Applications

- Training simulators
  - Improves realism
  - Decreases incorrect training

- Current uses
  - Tactical training
  - EMT training
Applications

● Multimedia

● Auditory displays
  - Enhance data visualization

● Telephony and Video conferencing
Applications

- Computer aided design
- Relay cues about environment design
- Preview room acoustics before construction
• Sound rendering and applications

• Details of propagation

• Our system: RESound
Propagation

- Simplest method:
  - Direct path between source and listener
  - Add echoes with post-process filter

- Fast

- Widely used
Propagation

- However
  - Not physically based
  - Spatialization incorrect
  - Echoes do not match environment
Propagation

• Acoustic simulations
  • Numerical
    – Solves acoustic wave equation
    – Slow, but getting faster [Raghuvanshi et al. 2009]
  • Geometric
    – High frequency approximation
    – Very fast – interactive
    – Models sound as ray
Propagation

- Specular reflection
  - Mirror-like reflections
  - Reflections decrease amplitude
  - Longer paths, longer delays
Propagation

- Specular reflection
  - Mirror-like reflections
  - Reflections decrease amplitude
  - Longer paths, longer delays
  - Often many reflection paths
Propagation

- Diffuse reflection
- Scattering reflections
Propagation

- Diffuse reflection
- Scattering reflections
- Scattered sound reaches listener
Propagation

• **Diffraction**
  - Sound 'bends' around corners
  - Can change phase
Propagation

• Diffraction
  • Sound 'bends' around corners
  • Can change phase
  • Often many diffraction paths
Propagation

- Combine
  - Direct
  - Specular
  - Diffuse
  - Diffraction
- Early contributions
  - 4-5 recursions
Propagation

- Reverberation
  - Late contributions
  - Impulses decays over time
  - Hundreds of recursions
  - Gives 'feel' of the room
Propagation

• Specular reflections
  • Image-source method [Allen et al. 1979]
  • From source
  • Reflect against all scene triangles
    - Creates image-sources
    - Is listener visible
  • Reflect image sources
    - and so on...
Propagation

- However
  - Very compute intensive
  - Need to accelerate

- Graphic acceleration
  - Remove non-visible triangles

- Sound acceleration
  - Remove non-reflecting triangles
Propagation

- Accelerated by
  - Ray tracing [Vorlander 1989]
  - Beam tracing [Funkhouser et al. 1998]
  - Frustum tracing [Lauterbach et al. 2007]
  - And others...

- Often require precomputation
  - Non-moving source
Propagation

- Diffuse reflections
  - Often modeled by ray tracing [Dalenbaeck 1996]
  - Radiosity [Siltanen et al. 2004]

- Compute intensive
  - Fixed source and receiver
  - No scene movement
Propagation

- Diffraction
- Added to
  - Beam tracing [Tsingos et al. 2001]
  - Ray tracing [Stephenson et al. 2007]
  - Frustum tracing [Taylor et al. 2009]
  - Image source [Shroeder et al. 2009]
Propagation

- Reverberation
- Ray tracing
  - Slow, accurate [Hodgson 1990]
- Statistical
  - Fast, some error [Savioja et al. 1999]
• Sound rendering and applications

• Details of propagation

• Our system: RESound
RESound

- Simulates all mentioned effects
- Interactive update rates
- Dynamic scenes
- Handles propagation and output
- Given input sound + environment
  - Propagates sound through environment
  - Renders signal at receiver's position
RESound

System overview
RESound

- Early contributions by simulation
  - Specular + diffraction
  - Diffuse reflection
- Late contributions by statistics
- 3d audio output
RESound

- Unified engine
  - Frustum tracing
  - Ray tracing
- Ray primitive
- Single acceleration structure
  - Bounding Volume Hierarchy
  - Allows dynamic scenes
  - Fast ray tracing
RESound

- Scene acceleration hierarchy
  - Bounding Volume Hierarchy [Lauterbach et al. 2006]
    - Fast construction times
    - Allows interactive visual ray tracing
    - Allows dynamic scene changes
  - Can accelerate frustum and ray tracing
• Specular + diffraction
  • Frustum tracing
  • Volumetric, finds most paths
  • Dynamic scenes
  • Fast

• Diffuse
  • Ray tracing
  • Shares scene structure
  • Dynamic scenes
  • Fast
RESound

- Frustum tracing
- Specular reflection
RESound

- Frustum tracing
  - Specular reflection
  - Frustum is bounded by rays
RESound

- Frustum tracing
  - Specular reflection
  - Check if receiver is inside bounded volume
RESound

- Frustum tracing
  - Specular reflection
  - Bounding rays can be reflected
RESound

- Frustum tracing
  - Specular reflection
  - Sound path is linear combination of rays
RESound

- Diffraction
  - Covers more area
  - Allows smooth transitions
    - Fades out
RESound

- Diffraction
  - Covers more area
  - Allows smooth transitions
    - Fades out

- First step
  - Find diffracting edges
RESound

- Frustum tracing
- Edge diffraction
RESound

- Frustum tracing
  - Edge diffraction
- From source
  - Trace many frusta
RESound

- Frustum tracing
  - Edge diffraction
- Receiver is hidden from source
RESound

- Frustum tracing
  - Edge diffraction
  - But diffracting edge is visible
RESound

- Frustum tracing
- Edge diffraction
- Create diffraction frustum
RESound

- Frustum tracing
  - Edge diffraction
  - Diffracting sound reaches the receiver
RESound

- Diffuse reflections
  - Uses ray tracing

- Collection sphere
  - Same size as listener's head (0.3 m)
  - Record rays that hit collection sphere
RESound

- Ray tracing
- Diffuse reflection
RESound

- Ray tracing
  - Diffuse reflection
  - Shoot rays from source
RESound

- Ray tracing
  - Diffuse reflection
  - Rays diffusely reflect
RESound

- Ray tracing
  - Diffuse reflection
  - Some rays hit this collection sphere
RESound

• Update stronger paths more often:

• Three simulations
  • Frustum tracing (first order, 1 thread)
  • Frustum tracing (third order, 7 threads)
  • Ray tracing, 200k rays (third order, 7 threads)
RESound

- From 3 simulations
- Now have impulse response of:
  - Direct sound
  - Specular reflection
  - Diffuse reflection
  - Edge diffraction
RESound

- Audio output
  - Reverberation
  - 3d sound rendering
  - Dynamic scenes
RESound

- Reverberation
  - Need to fill in late contributions
  - Use Eyring model [Eyring 1930]
  - Statistically estimate sound decay

- Combing impulse responses
  - Frustum + frustum + ray tracing
RESound

- Reverberation
  - Fit curve to impulse response
  - Estimate time for signal to decay to 0.001% ($RT_{60}$)
  - Create reverberation filter with sound system
RESound

- HRTF is expensive
  - Three impulse responses
    - $1^{st}$ order frustum tracing
    - $3^{rd}$ order frustum tracing
    - $3^{rd}$ order ray tracing
  - Compute only for $1^{st}$ order frustum tracing
  - Other impulses use simple convolution
RESound

- Dynamic scenes
  - Impulse response may change drastically
  - Can cause artifacts (clicking)
- Restrict motion speed
- Crossfade audio frames
Results

- Test scenes
Results

- Open scenes
  - Many triangles visible
  - Many reflections
Results
Results

• Reverberation
  • Begin with 6m cathedral
  • Dynamically expand cathedral to 30m
  • With reverb and without
Results
Results

● Limitations
  ● Must shoot many rays for diffuse reflections
  ● Certain diffraction paths may not be found
  ● Frustum tracing is approximate visibility
    – May miss some paths
Results

- Specular + diffuse + diffraction components
  - Uses unified representation: ray
  - Single acceleration structure
- Interactive rates on multi-core PC
- Handles large scenes
- Moving source and listener
- Scene can be dynamic
Related and Future Work

● Conservative frustum tracing [Chandak et al. 2009]

● GPU acceleration

● Robust diffraction
  ● Conservative diffraction region
  ● From region visibility – advanced diffraction
Acknowledgements

• Nikunj Raghuvanshi and Paul Calamia for helpful advice

• Sponsors
  – ARO
  – NSF
  – DARPA/RDECOM
  – Intel
  – Microsoft
Thanks!

Project website
http://gamma.cs.unc.edu/Sound/RESound/