Interactive Sound Propagation and Rendering for Large Multi-Source Scenes

Author: CARL SCHISSLER and DINESH MANOCHA

University of North Carolina at Chapel Hill

Presenter: Andrew Kim

Review of Last Week: Real-Time Polygonal-Light Shading with Linearly Transformed Cosines

- Rendering equation of Polygonal Light involves spherical integration
- Accurate spherical integration of BRDF is not possible
- Use of Linearly Transformed Cosines approximates various shapes of BRDFs with low complexity

Author: Eric Heitz
Presenter: In Yeong Cho

Table of Contents

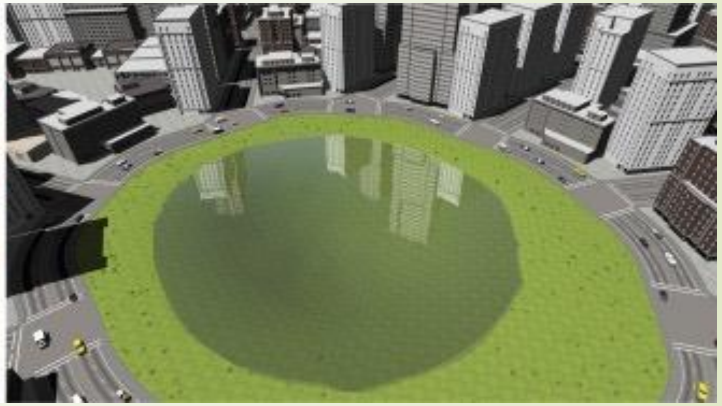
Video

- https://www.youtube.com/watch?v=0QjBUwIrD98

Introduction

- Visual complexity of scenes in media such as games are increasing
- Rendering sounds at such rate remains a big challenge
  - Large number of sound sources
  - Large number of objects
  - Simulating Acoustic effects





Goal / Problems

- Goal:

  Render **large number of sounds** in a **complex scene** at an **interactive rate**

- Problems:
  - Simulating reverberation
    - Sounds reaching the listener after a large number of reflections
    - Computationally expensive
  - Complexity of sound propagation algorithm
    - increases linearly as the number of sound sources increase
  - Real-time audio rendering
    - Hundreds of sources creates thousands of paths

Solutions

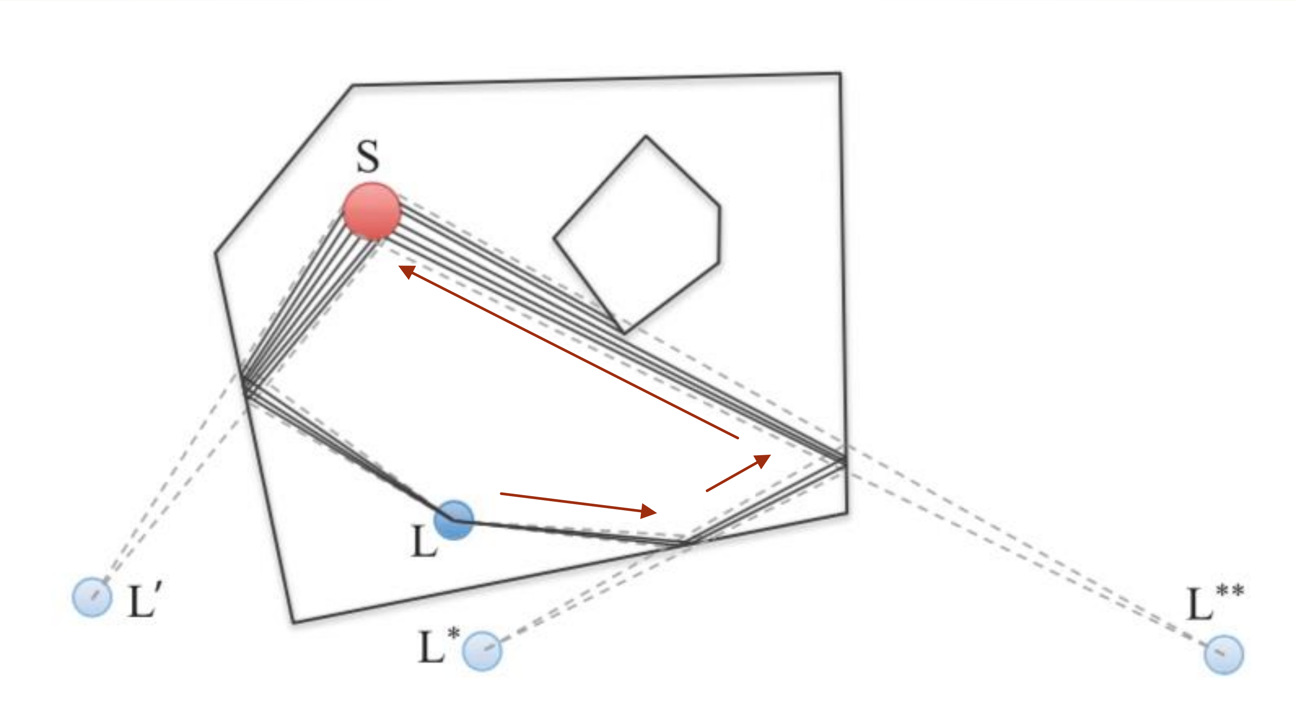
- 1. Acoustic Reciprocity for Spherical Sources
- 2. Source Clustering
- 3. Hybrid Convolution Rendering

Solutions

- **1. Acoustic Reciprocity for Spherical Sources**
- 2. Source Clustering
- 3. Hybrid Convolution Rendering

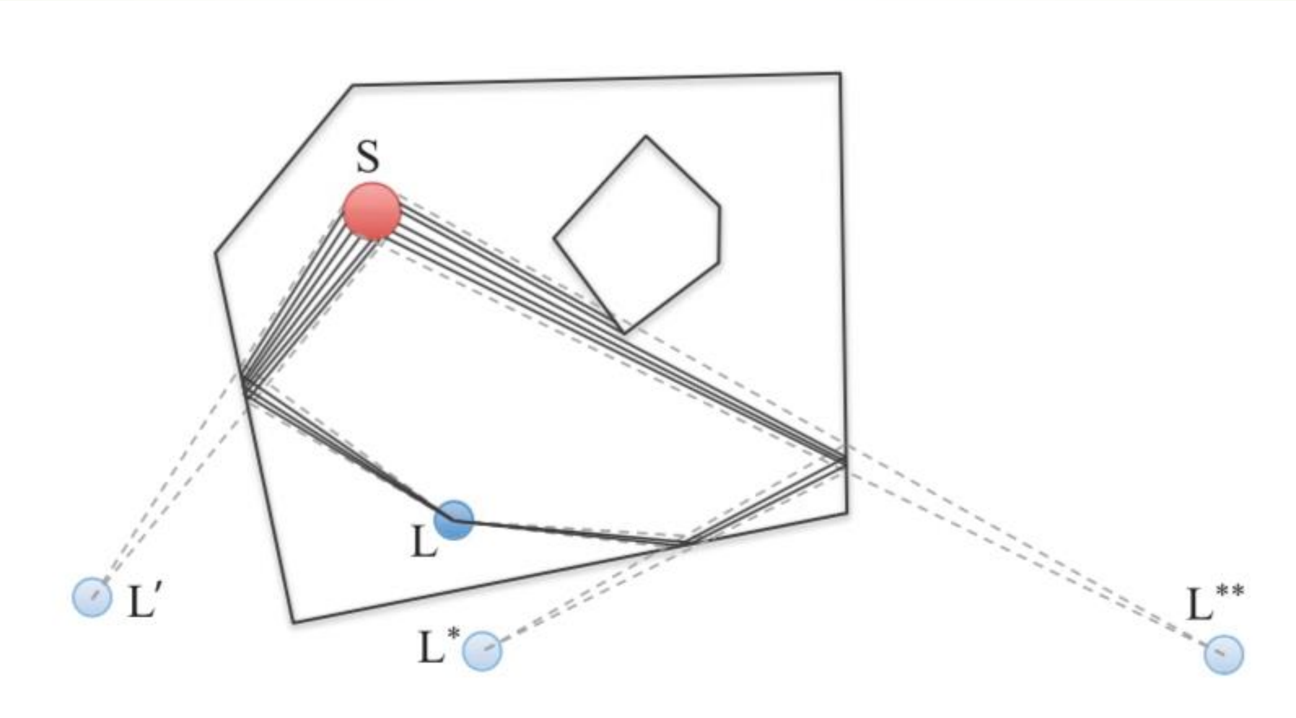
1. Acoustic Reciprocity for Spherical Sources

- Acoustic Reciprocity: **Backwards Ray Tracing**
  - Instead of the source, rays are traced from the listener
  - No longer linearly dependent on the number of sound sources



1. Acoustic Reciprocity for Spherical Sources

- Spherical Sources: **Representing Sound Sources as Spheres**
  - Cone of rays are fired back to the image of the listener
  - Rays that are not occluded by the obstacles go into the diffuse cache



Solutions

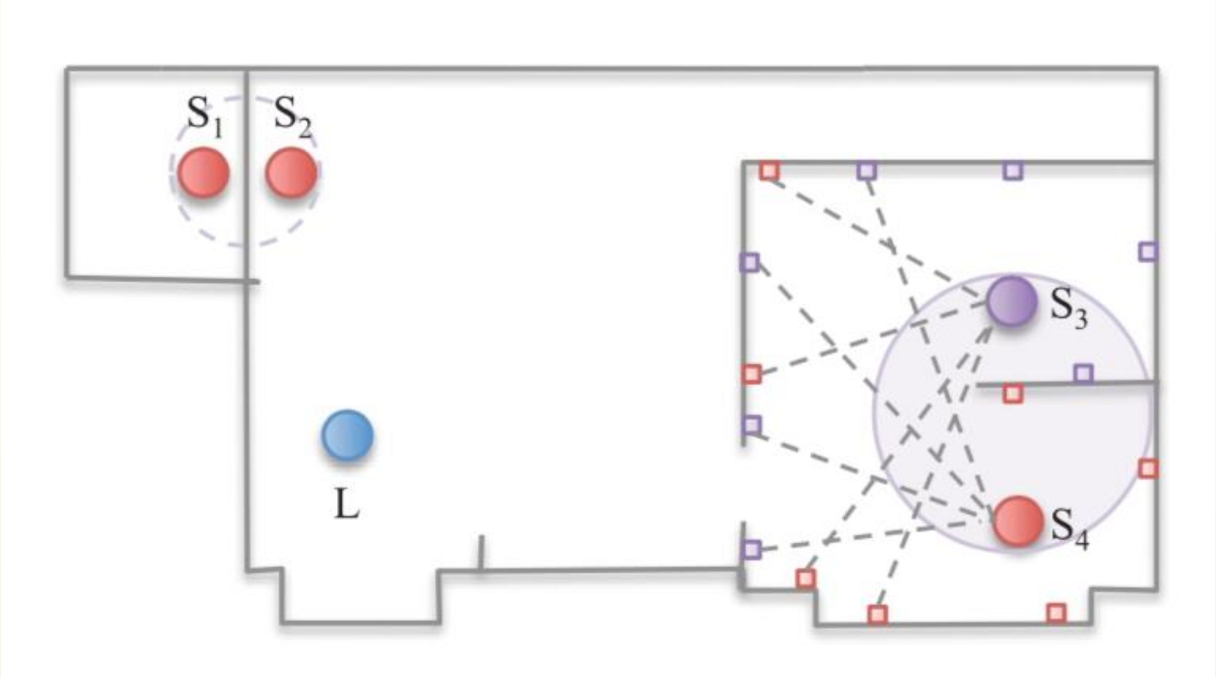
- 1. Acoustic Reciprocity for Spherical Sources
- **2. Source Clustering**
- 3. Hybrid Convolution Rendering

2. Source Clustering

- Sounds far away from the listener and are close to each other are 'clustered'
- Clustered sounds are treated as one spherical sound source

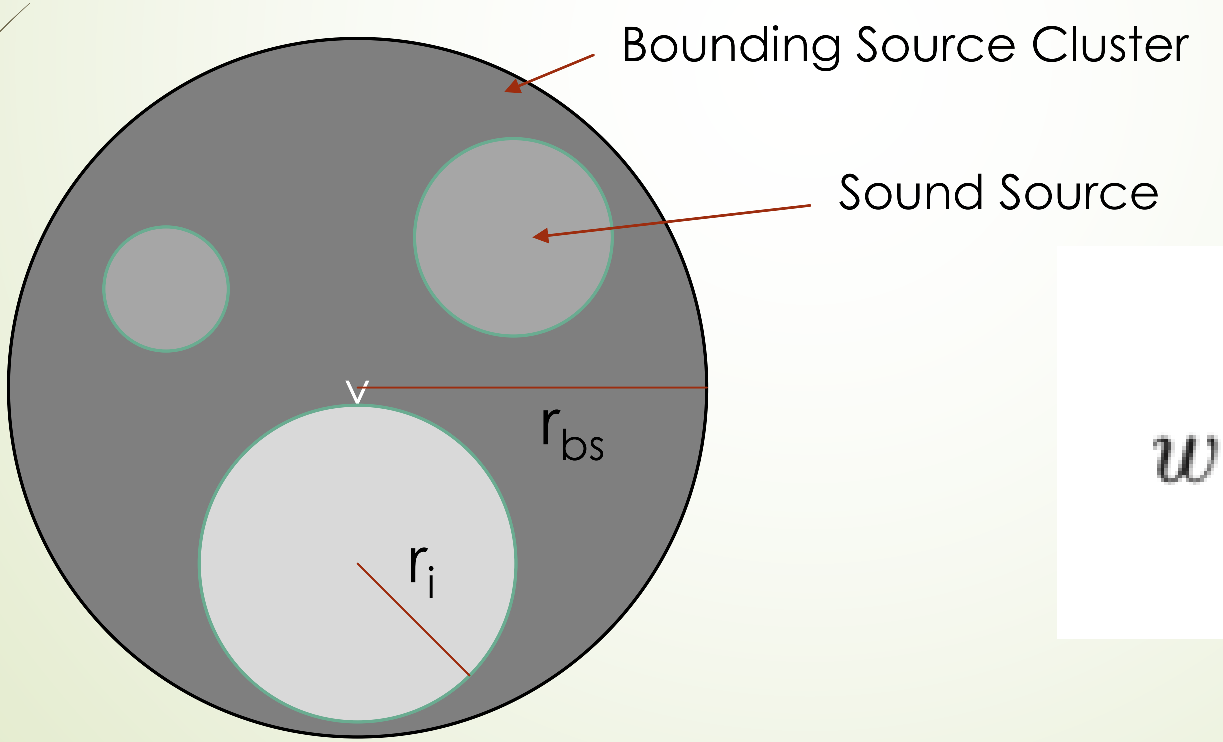
2. Source Clustering

- Clustering considers obstacles between the sound sources
- Rays are traced around the sound sources to see if the sources reside in the same acoustic space



2. Source Clustering

- Clustered sound source use large detection sphere, which may result in too much sound energy for source with small radii
- Normalization factor = (area of sound source silhouette) / (area of cluster silhouette)



$$w = \frac{\pi r_i^2}{\pi r_{BS}^2}.$$

Solutions

- 1. Acoustic Reciprocity for Spherical Sources
- 2. Source Clustering
- **3. Hybrid Convolution Rendering**

3. Hybrid Convolution Rendering

- A method to speed up the simulation of **Doppler Effect**
- Doppler Effect: Sound source that move towards or away from the listener generate different frequencies in relation to the velocity.

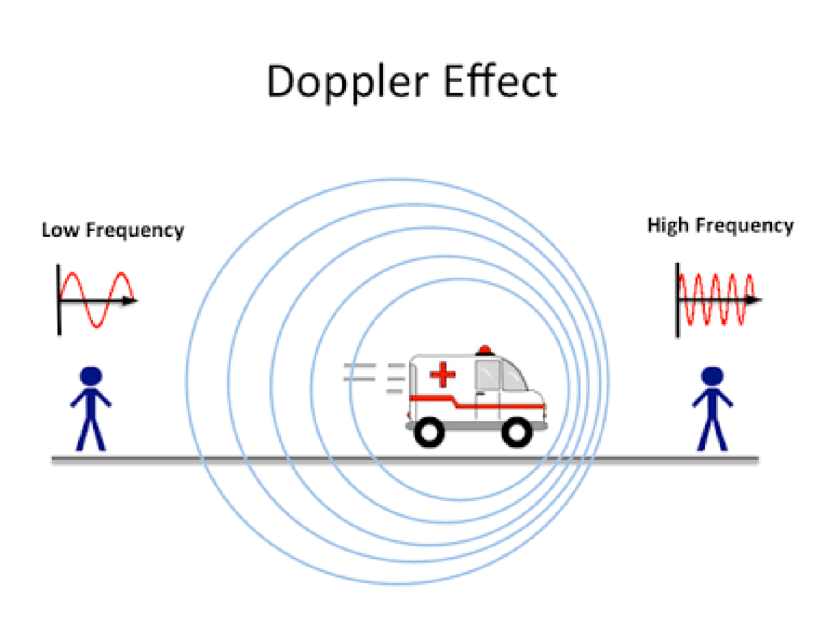


Image Source: https://www.quora.com/Why-does-the-Doppler-effect-happen

3. Hybrid Convolution Rendering

- Methods to render Doppler Effect:
  - **Fractionally interpolated delay lines**
    - Accurate rendering of Doppler Effect
    - Becomes expensive as the number of sound paths increases
  - **Partitioned frequency-domain convolution**
    - Handles large amount of Doppler Effects
    - Not an accurate simulation

Image Source: https://www.quora.com/Why-does-the-Doppler-effect-happen

3. Hybrid Convolution Rendering

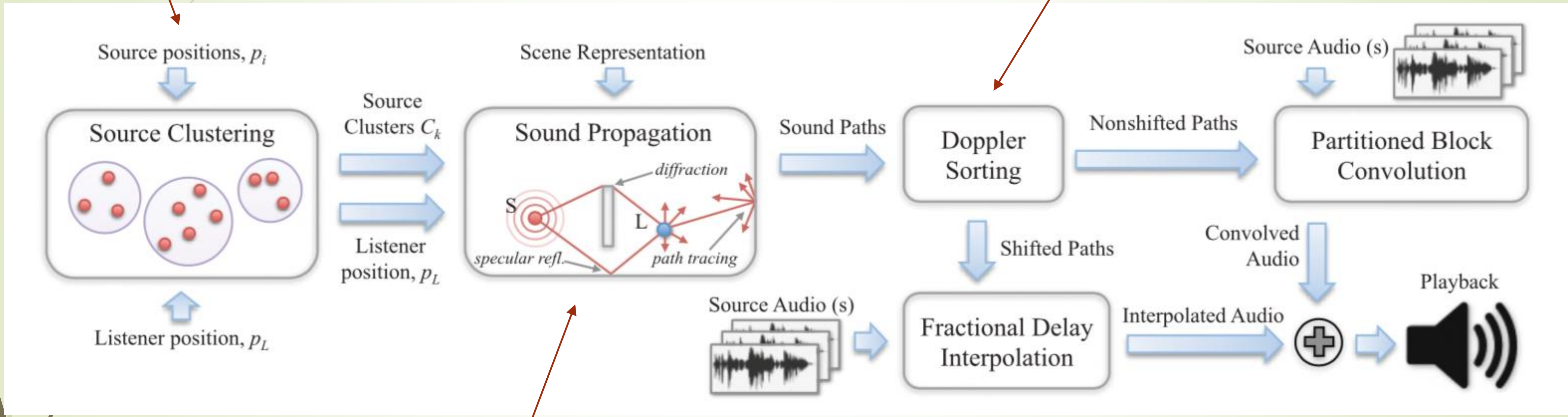
- Hybrid Convolution Rendering: **Combines the two methods**
- Doppler shift amount is calculated for each sound paths
- If (Doppler shift amount) > (threshold):
  - Use **Fractionally interpolated delay lines**
- If (Doppler shift amount) < (threshold):
  - Use **Partitioned frequency-domain convolution**
- If too many use of **Fractionally interpolated delay lines:**
  - Sort by decreasing amount of Doppler shift amount and sound path intensity

Image Source: https://www.quora.com/Why-does-the-Doppler-effect-happen

Overall Pipeline

- Source Clustering

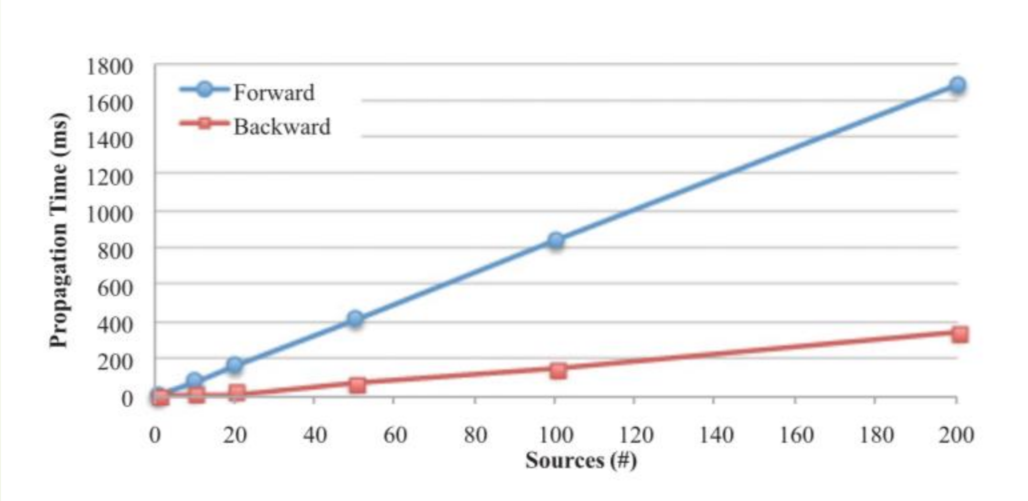
- Hybrid Convolution Rendering



- Backwards Ray Tracing
- Spherical Sources

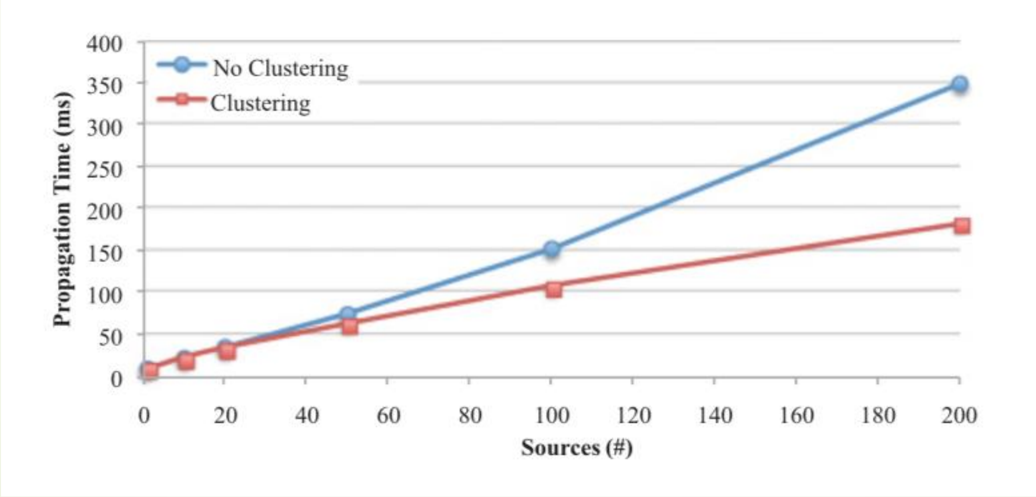
Results

- Backward ray tracing is 4.8 times faster than forward ray tracing
- Still has linear complexity



Results

- Clustering reduces the number of sources by around a factor of 2 on average
- Clustering efficiency is increased as more sources are far away from the listener



Limitations

- Since it is based on ray-tracing, it cannot accurately simulate low frequency sounds
- Assumption that all scene primitives are larger than the wavelength
- Representation of sound sources as spheres may not work well in some situations

Summary

- Render **large number of sounds** in a **complex scene** at an **interactive rate** using:
  - 1. Acoustic Reciprocity for Spherical Sources
    - Backwards Ray Tracing – Rays from listener to sound sources
    - Spherical sound source – Allows smooth interpolation
  - 2. Source Clustering
    - Clustered when sound sources are far away from the listener
    - Clustered when sound sources are close to each other with no obstacles
  - 3. Hybrid Convolution Rendering
- Successful at rendering large number of sound sources in a complex scene at an interactive rate