Applying Visibility Term between Clustered Sources to Improve Sound Source Clustering

Presenter: Andrew Kim

Original Paper: Interactive Sound Propagation and Rendering for Large Multi-Source Scenes

- 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

Author: CARL SCHISSLER and DINESH MANOCHA
University of North Carolina at Chapel Hill

Original Paper: Interactive Sound Propagation and Rendering for Large Multi-Source Scenes

- 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

Author: CARL SCHISSLER and DINESH MANOCHA
University of North Carolina at Chapel Hill

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

Source Clustering

- Implemented using "**octree**"
- For 'sources must be close together' criteria:
  - Sources are clustered if they are in the same node
- For 'sources must be far away from the listener' criteria:
  - Node is subdivided if sound sources in the node are too close to the listener

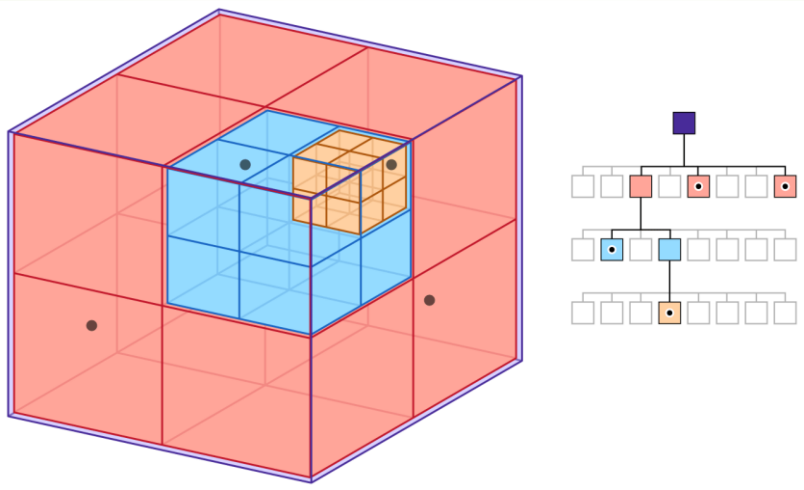
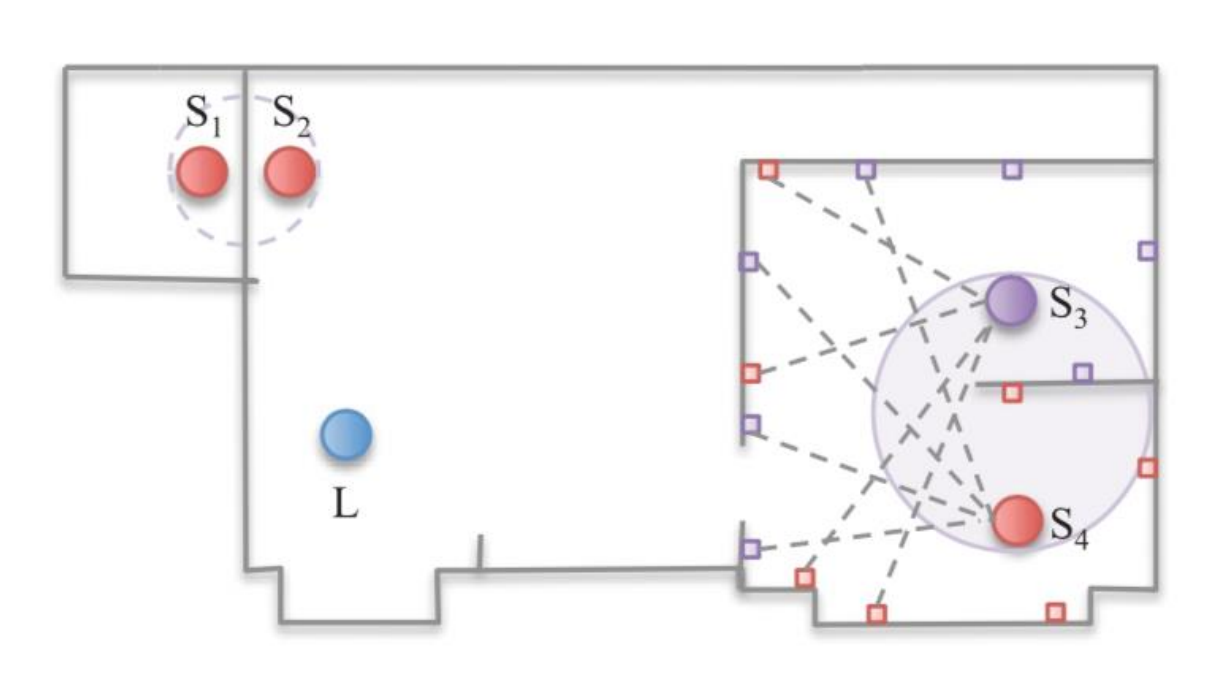


Image Source: https://developer.apple.com/documentation/gameplaykit/gkoctree

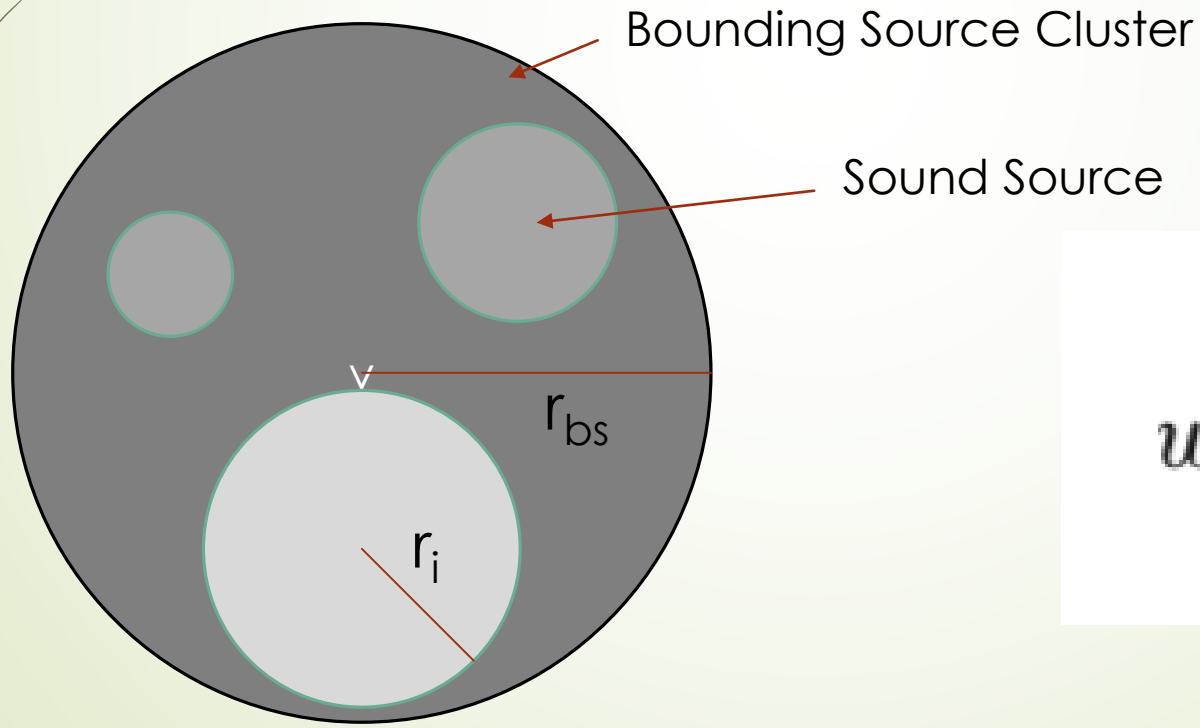
Source Clustering

- 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
- Given the number of rays that intersect, they are given visibility term **v** (ranging from 0 to 1).



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)



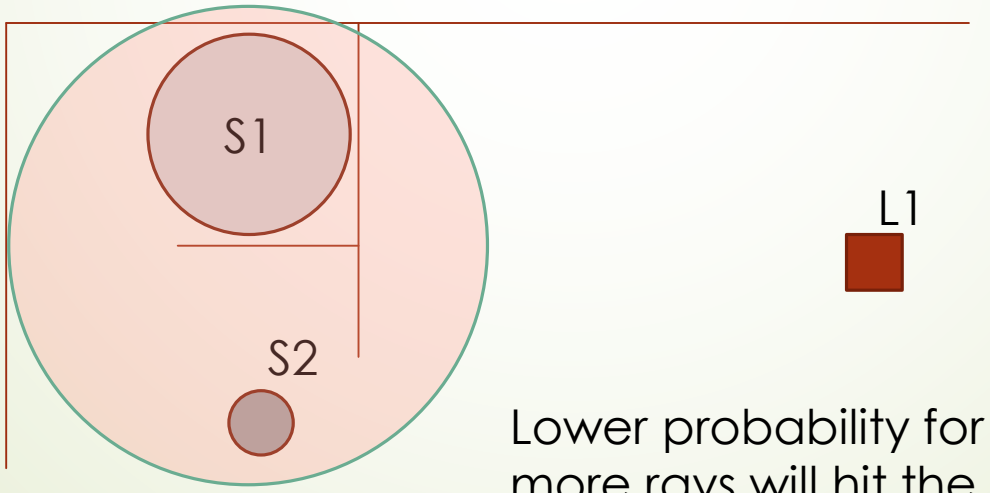
$$w = \frac{\pi r_i^2}{\pi r_{BS}^2}.$$

Goal / Problems

- Goal of the Project:

  To improve on the current clustering algorithm

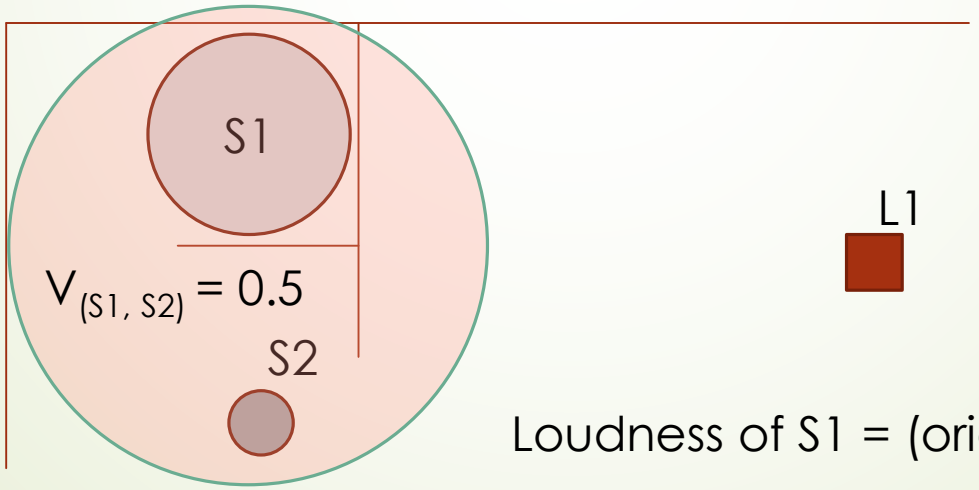
- Found Problems:
  - If large sound source is partially occluded and also clustered, it may result in an inaccurate simulation result



Lower probability for L1 to hear S1, but more rays will hit the cluster

Solutions

- Applying visibility term **v** during actual sound processing
  - Must path-find to check which sound source is further from the listener



Loudness of S1 = (original loudness) * $V_{(S1, S2)}$

Platform / Implementation

- **Unreal Engine 4 + Steam Audio**
  - Unreal Engine 4 for 3D Interactive Environment
  - Steam Audio for Sound Propagation
    - Geometric propagation (reflection, diffuse…) using backwards ray tracing (from listener to sound sources) included in the package
  - Implement "Improved Clustering Algorithm" using C++ in Unreal Engine Script

UNREAL ENGINE + STEAM® AUDIO