Adaptive Impulse Response Modeling for Interactive Sound Propagation

Author: CARL SCHISSLER and DINESH MANOCHA

University of North Carolina at Chapel Hill

Presenter: Andrew Kim

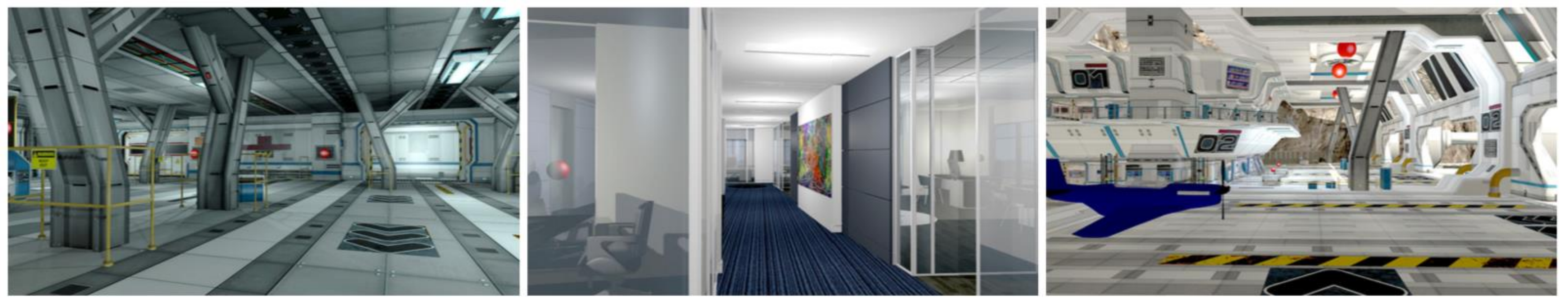
Table of Contents

Video

- https://www.youtube.com/watch?v=_7LYndtug2k

Introduction

- Realistic sounds are important in user's immersion in virtual environments.
- The computation of interactive sound propagation in a complex scene is expensive
  - The number of rays to trace increases as the order of reflection increases



Goal / Problems

- Goal:

  Render **impulse response (IR)** in a **complex scene** at an **interactive rate**

- Problems:
  - The complexity of geometric sound propagation algorithm depends on the computation of IRs
  - Choosing the appropriate IR length is a major issue in cutting off sounds that are inaudible to human ear

Solutions

- 1. Impulse Response Cache
- 2. Adaptive Impulse Response Length

Solutions

- **1. Impulse Response Cache**
- 2. Adaptive Impulse Response Length

1. Impulse Response Cache

- At frame n, uniform random rays are traced, generating a slightly different impulse response from that of previous frames
- Let's call this the **current frame's tracing output, $\tilde{h}_n^i$**

Current frame's tracing output

Previous frame's final IR sample

Final IR sample

$$H_n^i = \alpha \tilde{h}_n^i + (1 - \alpha)\, H_{n-1}^i$$

Contribution of current frame's IR

Contribution of previous frame's final IR sample

1. Impulse Response Cache

- The Impulse Response Cache has the final IR sample of the previous frame n-1
- Let's call this **previous frame's final IR sample**, $\boldsymbol{H_{n-1}^{i}}$

Current frame's tracing output

Previous frame's final IR sample

Final IR sample

$$H_n^i = \alpha \tilde{h}_n^i + (1 - \alpha)\, H_{n-1}^i$$

Contribution of current frame's IR

Contribution of previous frame's final IR sample

1. Impulse Response Cache

- **Contribution weight α** is distributed to these two different IR samples
- They are then added together to calculate the **final IR sample of current frame n, H_n^i**
- The **final IR sample H_n^i** is stored into the Impulse Response Cache for future IR calculation of the next frame, H_{n+1}^i

Current frame's tracing output

Previous frame's final IR sample

Final IR sample

$$H_n^i = \alpha \tilde{h}_n^i + (1 - \alpha)\, H_{n-1}^i$$

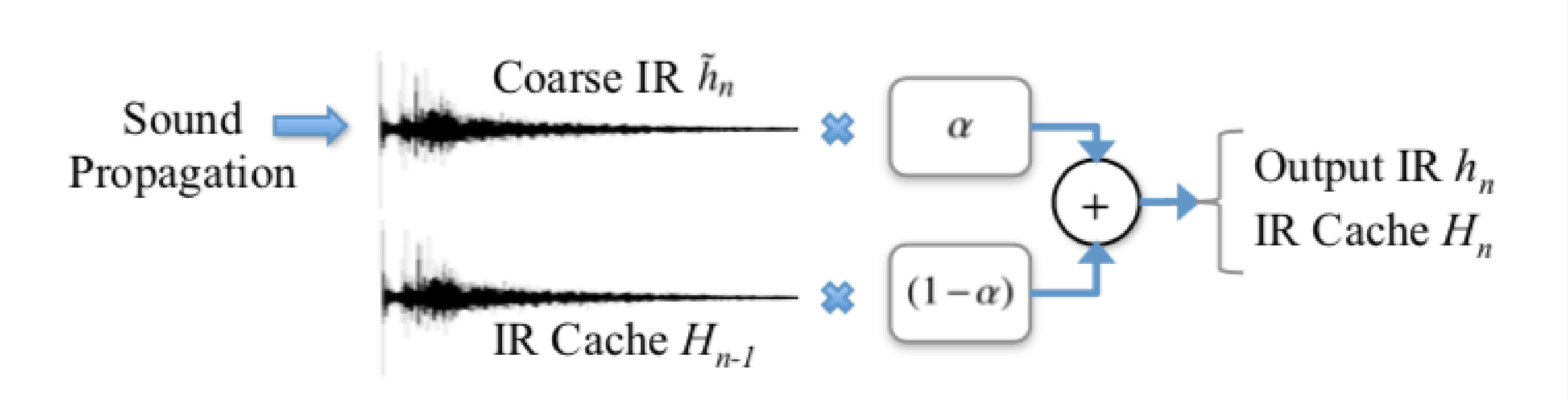
Contribution of current frame's IR

Contribution of previous frame's final IR sample

1. Impulse Response Cache

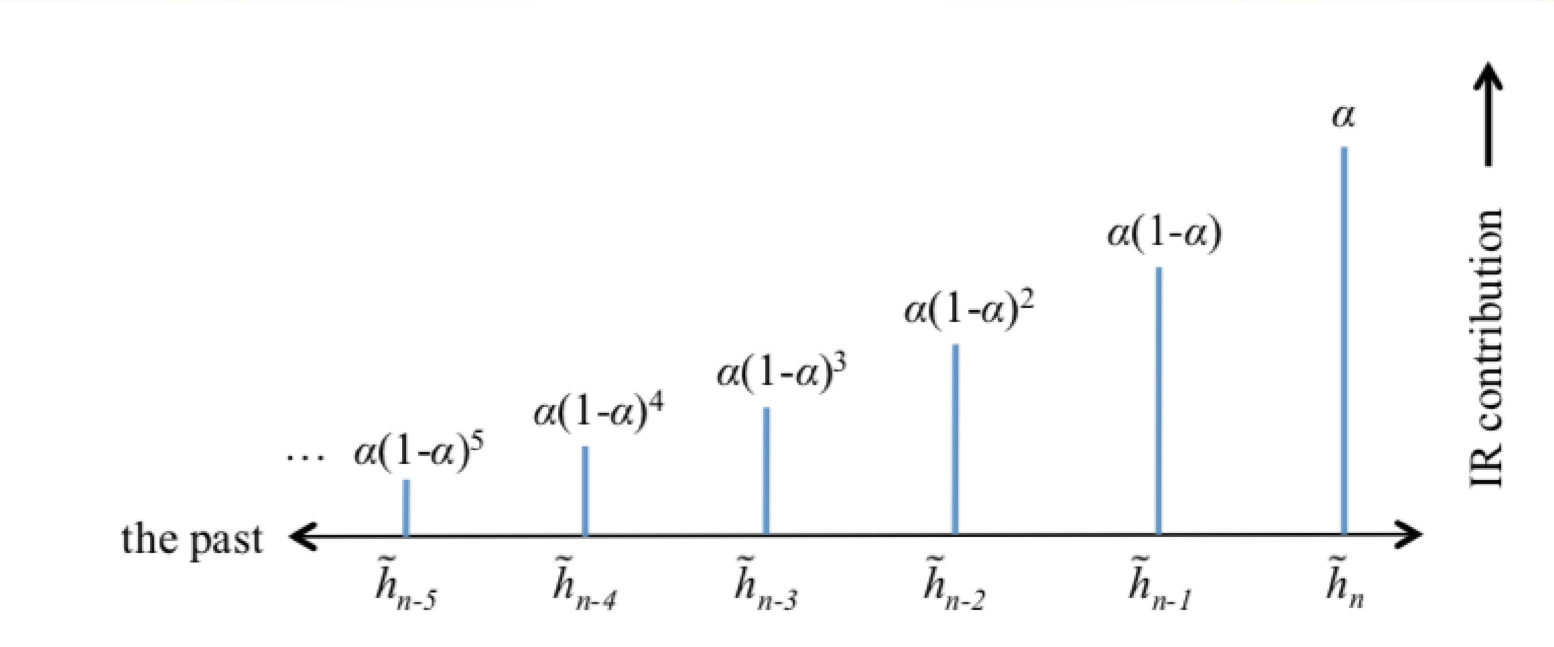
- Previous frame's final IR sample, H_{n-1}^{i}
- Current frame's tracing output, $\tilde{h}_{n}^{i}$
- **Contribution weight** $\boldsymbol{\alpha}$ is distributed to these two different IR samples

$$H_n^i = \alpha \tilde{h}_n^i + (1 - \alpha)\, H_{n-1}^i$$



1. Impulse Response Cache

- Rays from previous frames contribute to the calculation of IR – **better quality**
- Fewer rays are traced each frame – **faster computation time**



1. Impulse Response Cache

- **Contribution weight** $\boldsymbol{\alpha}$ has values between **0** and **1**
- Value closer to **1** means more contribution from current frame's tracing output
  - The system is more responsive to dynamic changes in the scene
- Value closer to **0** means more contribution from previous frame's final IR sample
  - The system benefits more from the cache but be less responsive

Current frame's tracing output

Previous frame's final IR sample

Final IR sample

$$H_n^i = \alpha \tilde{h}_n^i + (1 - \alpha)\, H_{n-1}^i$$

1. Impulse Response Cache

- **Determining Contribution weight $\boldsymbol{\alpha}$:**
  - The contribution of j[th] previous frame is $\boldsymbol{\alpha(1-\alpha)^{j}}$
  - If this value is less than some small value $\boldsymbol{\epsilon}$,
  - Solving this equation gives $\alpha = 1 - \epsilon^{\Delta t/\tau}$
  - τ is the filtering window

$$\alpha(1-\alpha)^{j} \leq \epsilon \longrightarrow \alpha = 1 - \epsilon^{\Delta t/\tau}$$

$$(j = \tau / \Delta t)$$

1. Impulse Response Cache

- **Determining filtering window τ:**
  - Smaller value of τ is chosen towards beginning of IR and a larger value towards the end
  - Delay time d
  - It is possible to use other function for τ

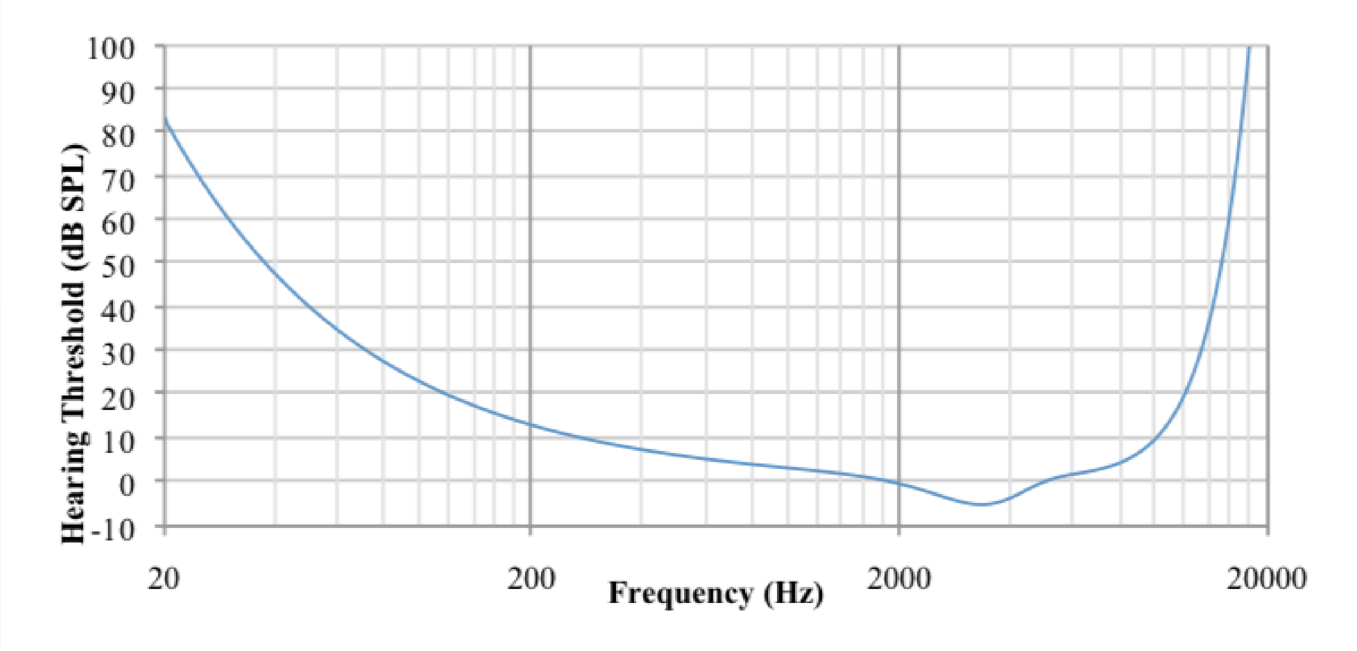
$$\tau = \max(\beta d, \tau_{min})$$

Solutions

- 1. Impulse Response Cache
- **2. Adaptive Impulse Response Length**

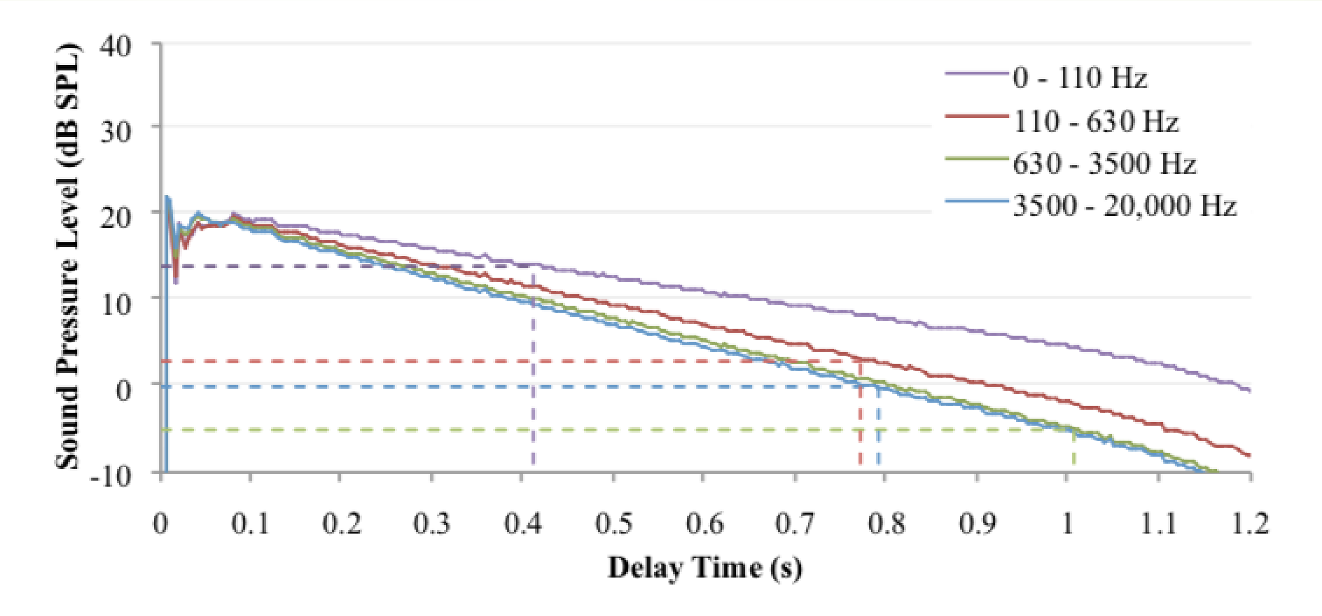
2. Adaptive Impulse Response Length

- **Psychoacoustic metric** is used to determine human audible IR length for each frequency band
- If IR length is greater than the threshold, rays are not traced further



2. Adaptive Impulse Response Length

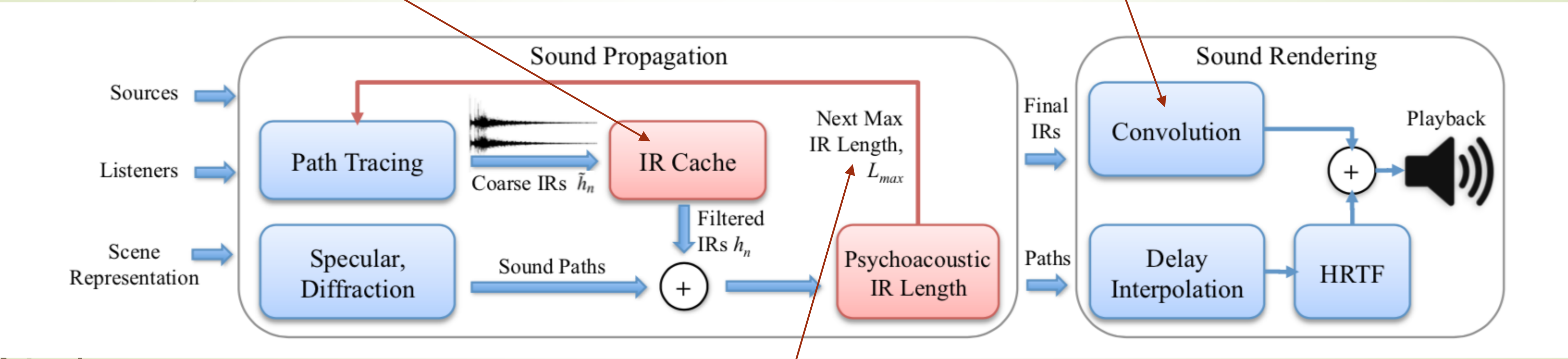
- However, simply tracing rays upto the threshold **L** artificially cuts IR length if listener enters a more reverberant space
- $L_{max} = L + \Delta L$
- The larger the value of ΔL, the quicker the system reacts to the change in IR length, at the expense of tracing further rays



Overall Pipeline

- Impulse Response Cache

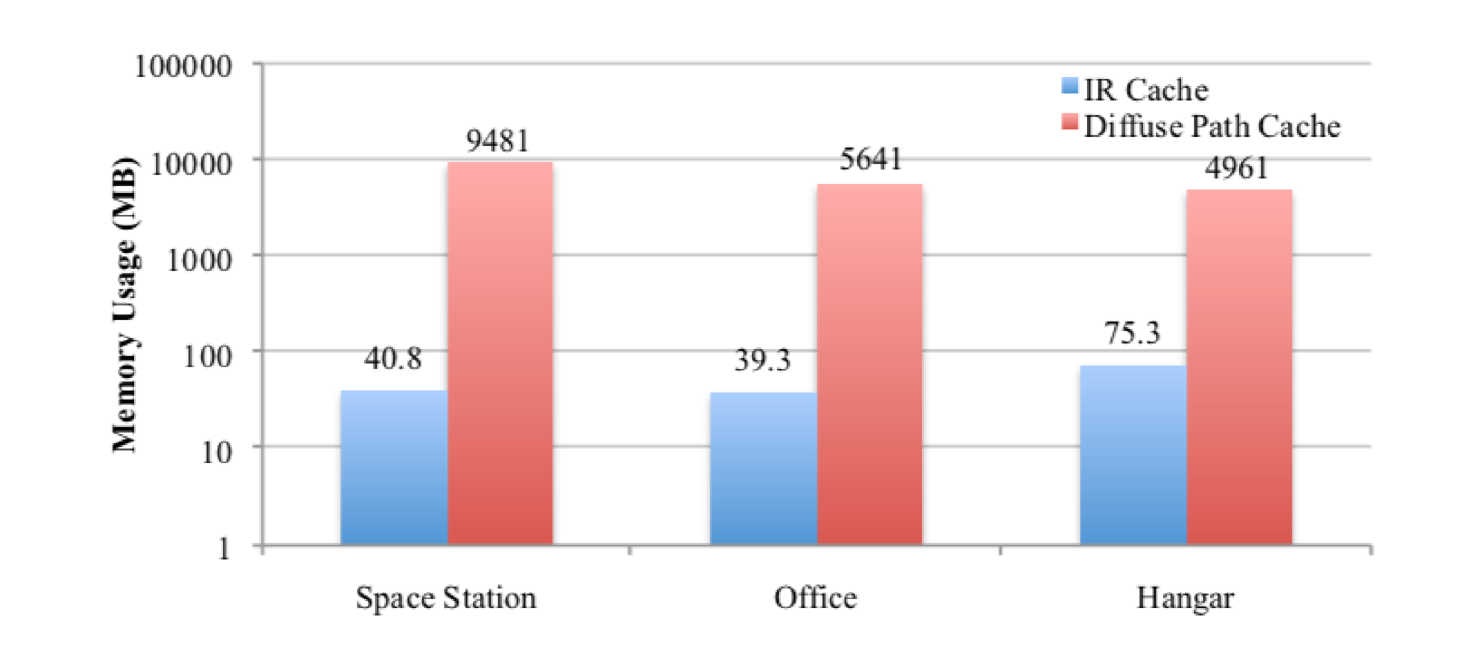
- Final Sound Rendering



- Adaptive Impulse Response Length

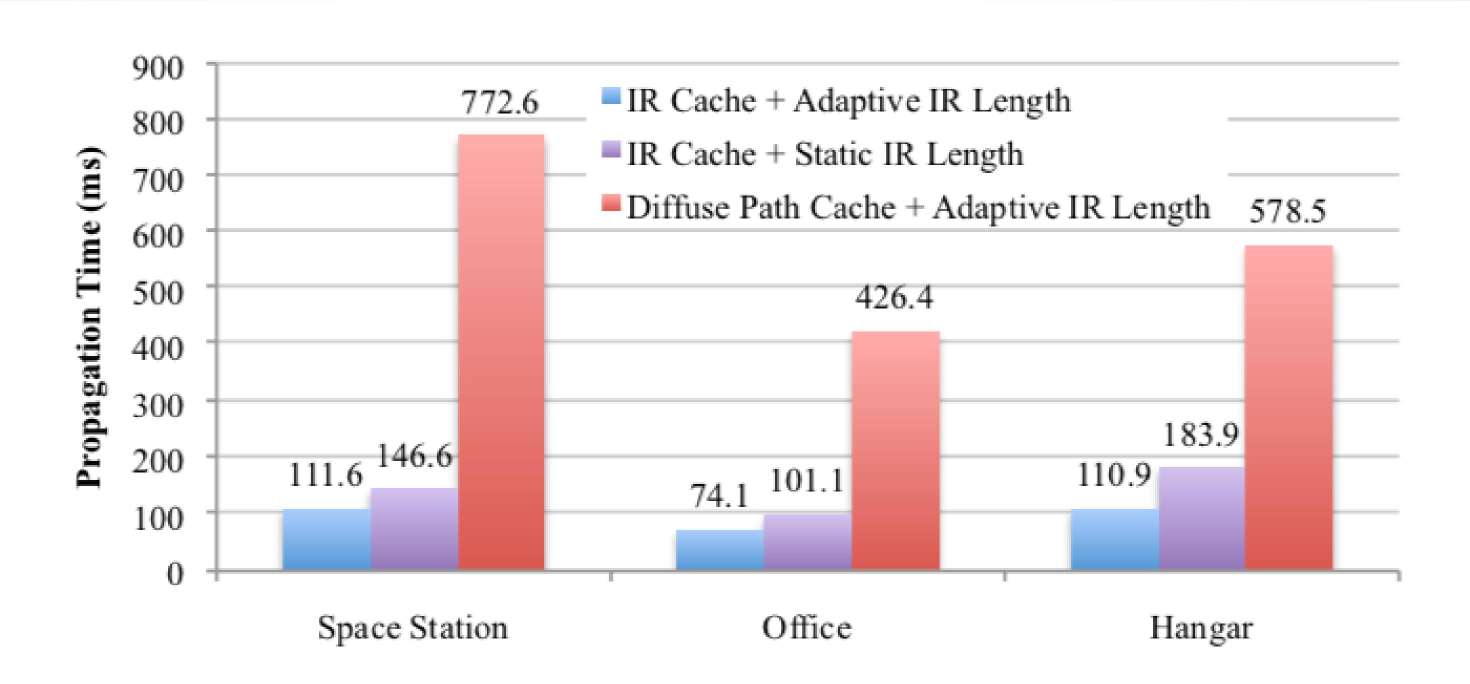
Results

- IR Cache uses **2 times less memory** than standard approach
- **Less rays are traced** due to the use of IR cache, increasing performance



Results

- Using Adaptive IR Length compared to Static IR Length provides **30%~60% speedup**



Limitations

- Since it is based on ray-tracing, it cannot accurately simulate low frequency sounds
- Maximum diffraction order was limited to 3, so some sound paths may be missed
- The psychoacoustic metric may not apply to all users as they have different hearing threshold

Summary

- Render **impulse response (IR)** in a **complex scene** at an **interactive rate**
  - 1. Impulse Response Cache
  - 2. Adaptive Impulse Response Length
- Successful in reducing computation overhead and memory usage with accurate estimate to ground-truth IR