



注釈に発表の原稿を記載しています。PDFリーダーで表示してご参照ください。

The manuscript of the presentation is described in the annotation. Please check with PDF reader.

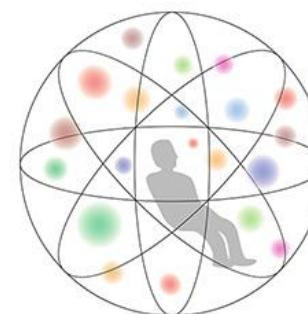
A Low Cost Method for Applying Acoustic Features to Each Sound in Virtual 3D Space Using Layered Quadtree Grids

Kenji Kojima (kenji-kojima@capcom.com)

Takahiro Kitagawa (takahiro-kitagawa@capcom.com)

CAPCOM CO., LTD.

CAPCOM



AES 2018 International Conference on
SPATIAL REPRODUCTION
~ Aesthetics and Science ~



2018, August 7th-9th
(pre-event on August 6th)
Tokyo, Japan

Tokyo Senju Campus, Tokyo Denki University
Senju Campus, Tokyo University of the Arts

Authors

- Kenji Kojima
 - 1st Author
 - Mainly proposed basis of theory
- Takahiro Kitagawa
 - 2nd Author, Speaker
 - Mainly conducted experimental evaluation



Contents

1. Introduction
2. Determining Sound Space Using LQG
3. Designing Propagation Using Rearrangement Method
4. Evaluation
5. Conclusion



Contents

1. Introduction
2. Determining Sound Space Using LQG
3. Designing Propagation Using Rearrangement Method
4. Evaluation
5. Conclusion



What is required for video game sound?

Realistic sound enhances player immersion

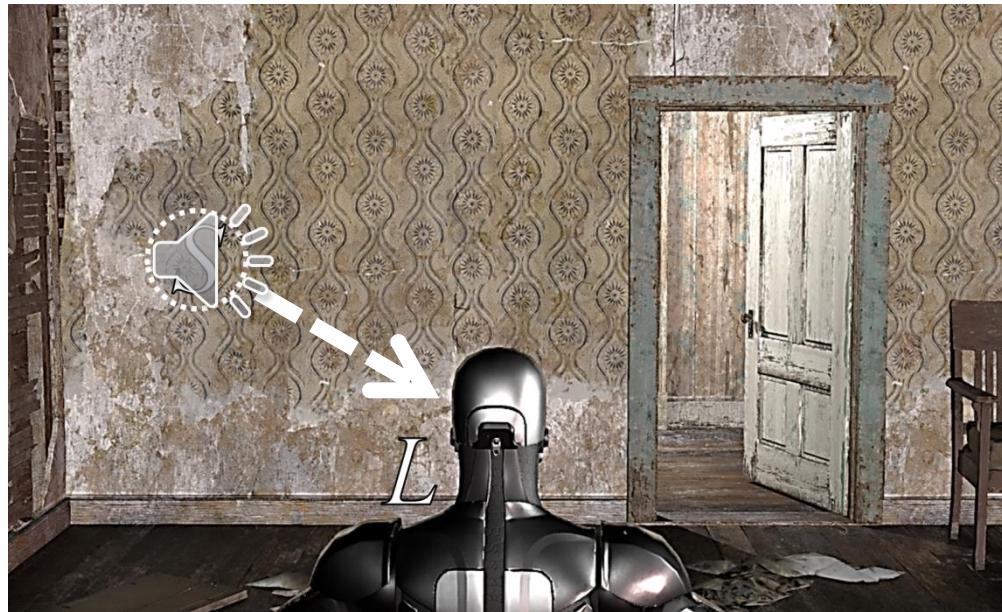


Imagine you hear a noise from the other side of the wall...



What is required for video game sound?

Realistic sound enhances **player immersion**



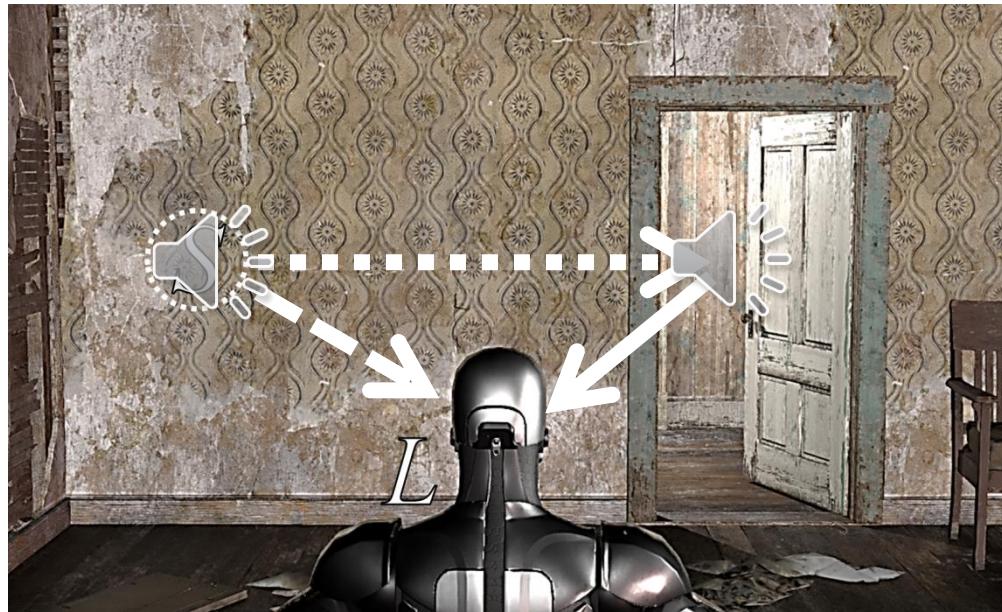
Imagine you hear a noise from the other side of the wall...

Does it sound **muffled**?



What is required for video game sound?

Realistic sound enhances **player immersion**

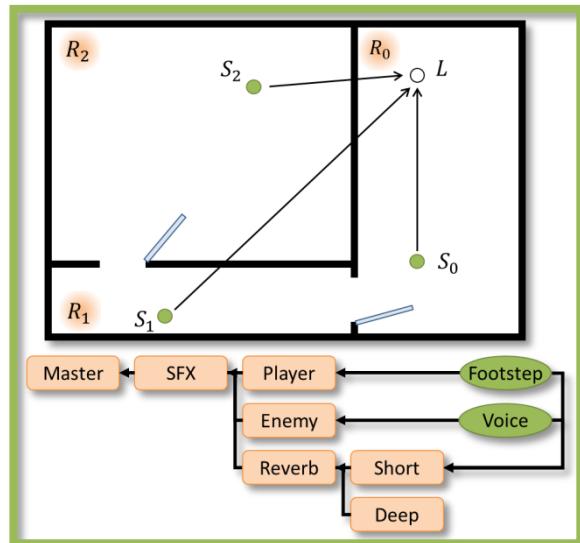
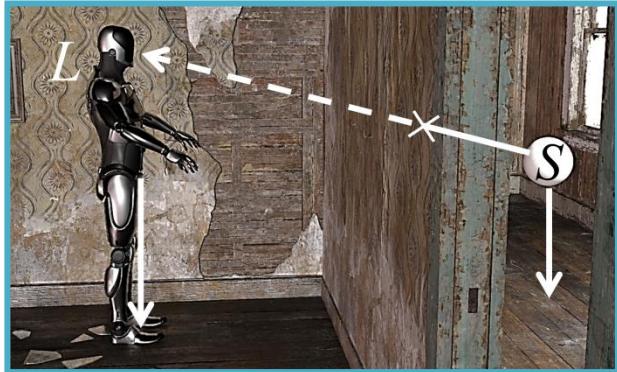


Imagine you hear a noise from the other side of the wall...

Does it sound **diffracted**?



How do we represent space with sound?



1. Determine sound spaces

- To determine **listener & source spaces**, **cast rays** down from sources and listener
- To find **obstacles** between listener & sources, **cast rays** from sources to listener

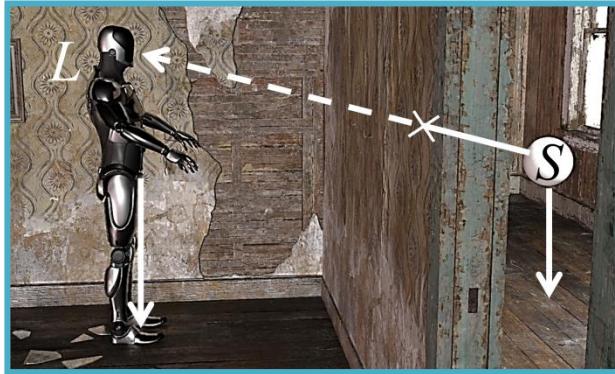


2. Represent using buses & effects

- Configure **buses** according to source type
- Send sources to **reverb buses** according to their spaces and obstacles

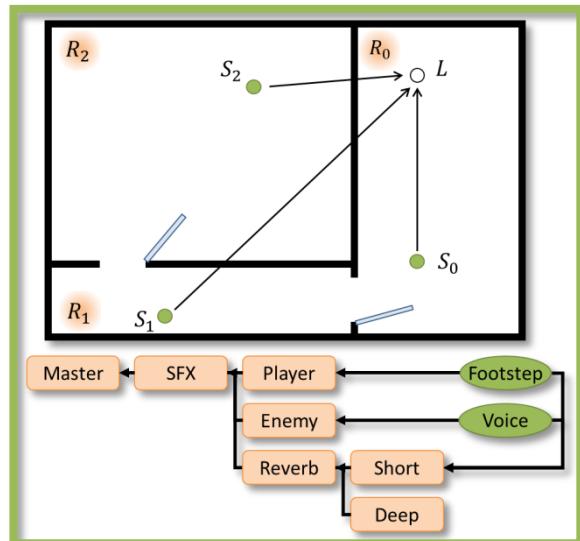


What issues do we face?



1. Determine sound spaces

- Casting rays, especially long ones, **take up lots of CPU resources**

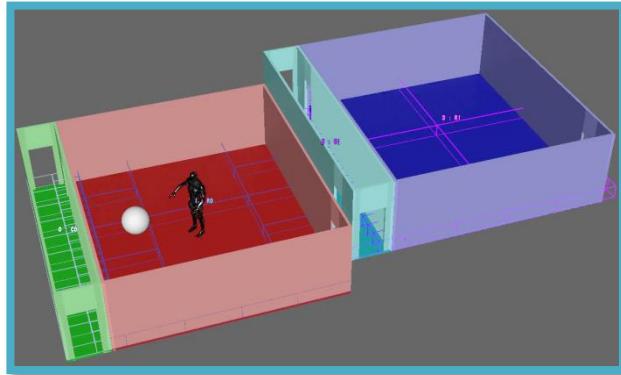


2. Representation using buses & effects

- Insufficient representation
 - No **intermediate** spaces represented
 - No **leaking** from connections
- Wasted CPU resources
 - **Individual processes** wasted on unheard sources

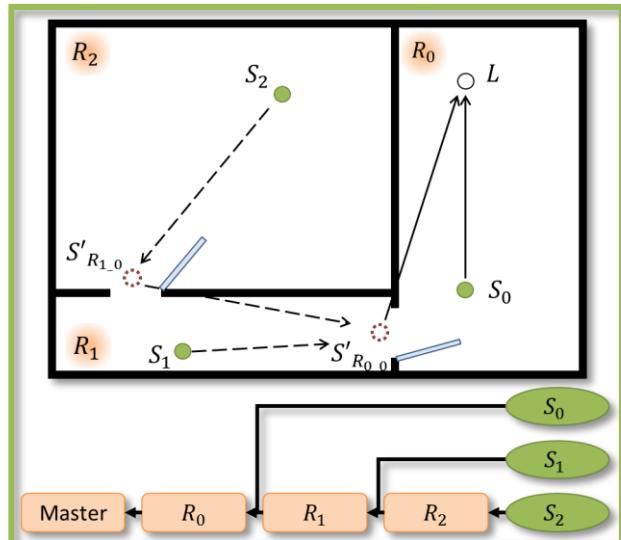


How can we solve these issues?



1. Layered Quadtree Grids (LQG)

- **Fast determination** of spaces without using rays



2. Sound source rearrangement

- Configured **buses** represent **spatial connections** in more detail



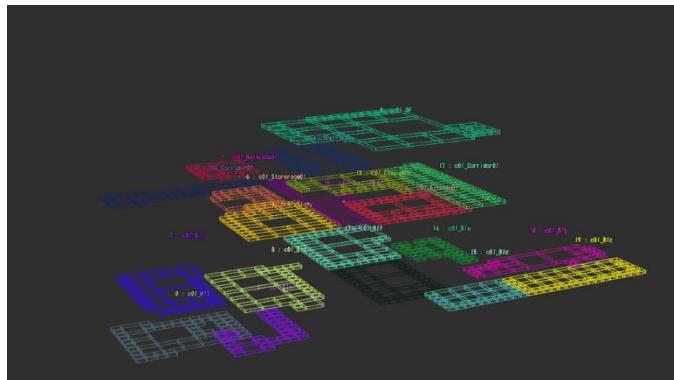
Contents

1. Introduction
2. Determining Sound Space Using LQG
3. Designing Propagation Using Rearrangement Method
4. Evaluation
5. Conclusion



Theory

How to quickly determine spaces without rays



Q. In **what space** are the listener and sources?

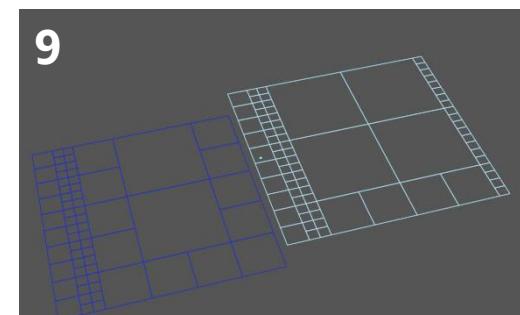
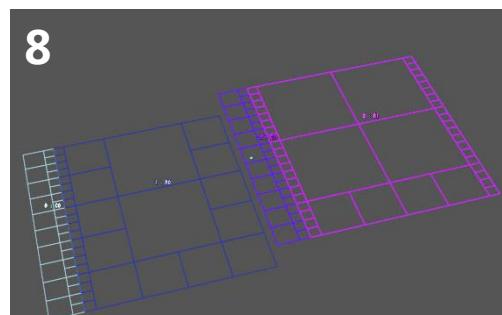
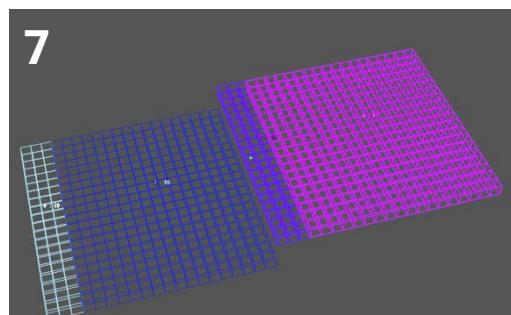
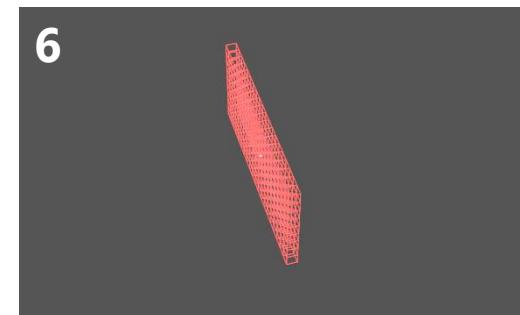
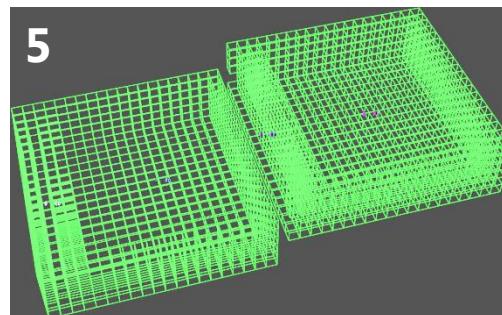
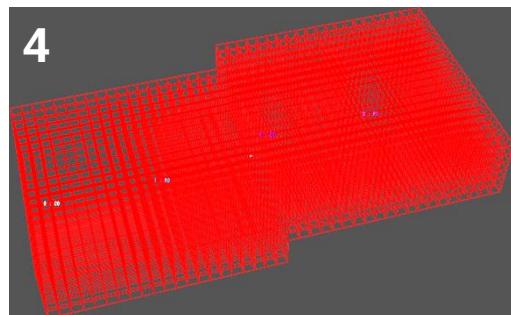
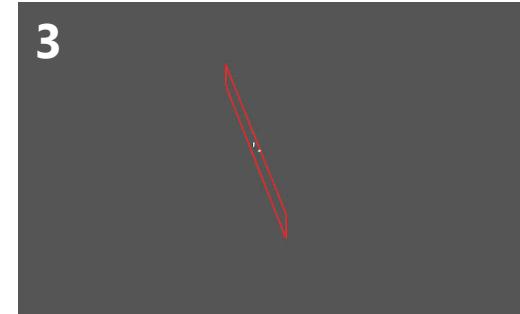
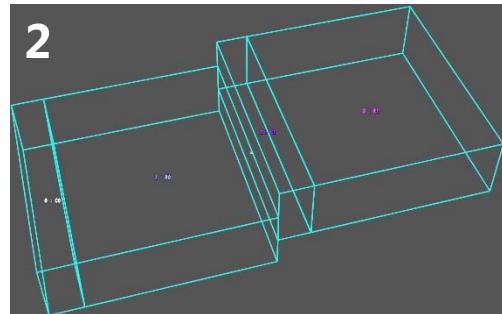
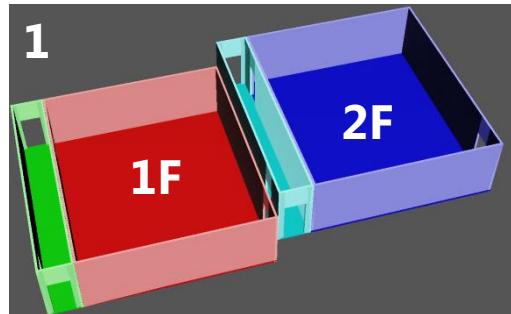
A. They are in the **1st floor kitchen**, ~~30 cm above the floor, in the corner of the room.~~

Simplification

Consider 3D spaces as **Quadtree Grids layered** at a fixed height

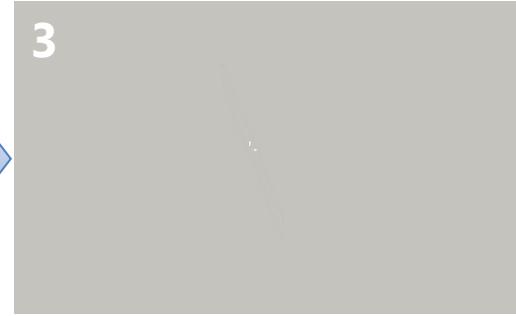
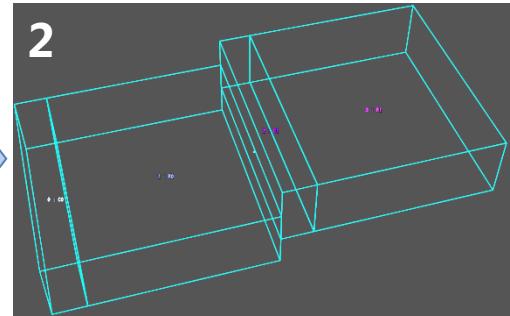
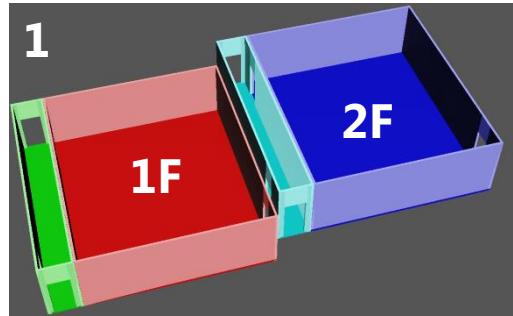
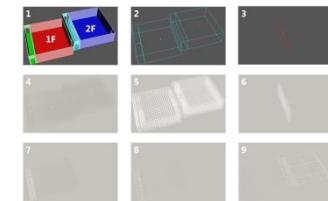


LQG Computing Algorithm





LQG Computing Algorithm



LQG Target Rooms

- Total: 4 Spaces
 - 1F
 - Corridor
 - Square room
 - 2F
 - Corridor
 - Square room

Bounding AABBs

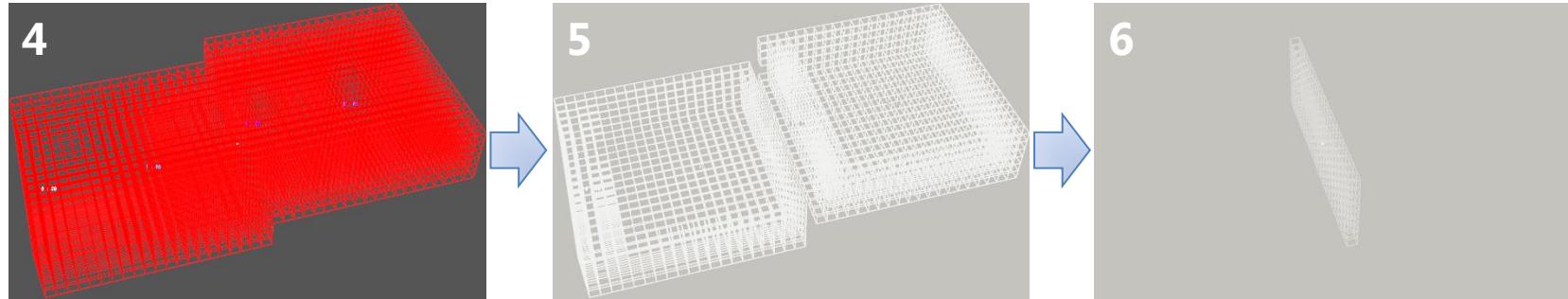
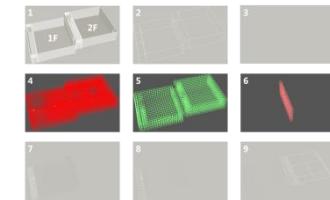
- Obtain the smallest AABBs covering each space of the target rooms

Overlapping Bounding AABB

- Obtain the overlapping Bounding AABB
- In this case, it's the boundary between 1F & 2F



LQG Computing Algorithm



Voxels Fill in Bounding AABBs

- These voxels have information on which space they belong

Voxels(1) Wrapping Target Surfaces

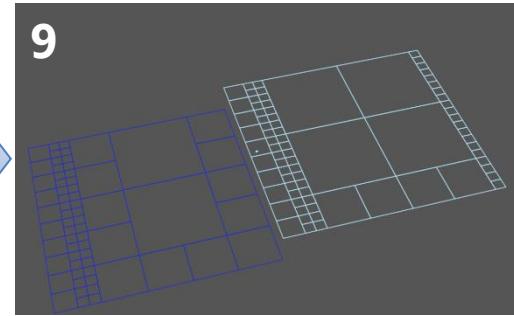
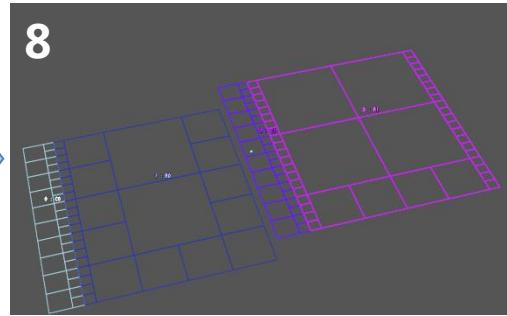
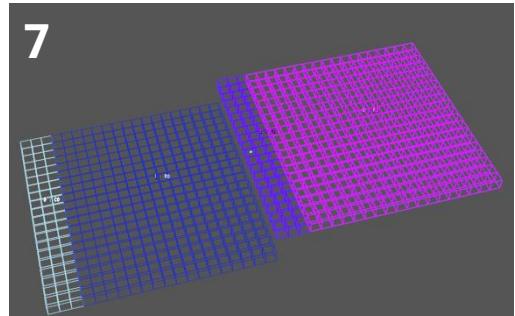
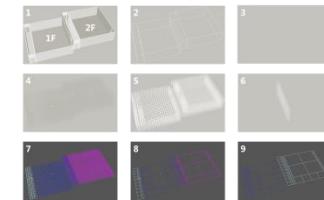
- Obtain the voxels that collide with the target (not the Overlapping Bounding AABB)

Voxels(2) Wrapping Target Surfaces

- Obtain the voxels that collide with the Overlapping Bounding AABB
- **Associate voxels with the spaces by casting rays since they're not predetermined**



LQG Computing Algorithm



Floor Grids

- Remove voxels that differ only in height except for the lowest

Quadtree Grids

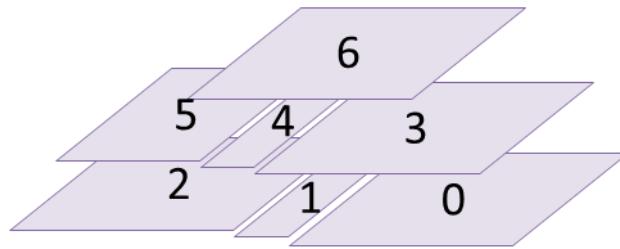
- **For faster searching,** convert the Floor grids into squares with the number of grids equal to 2^N

Layered Quadtree Grids

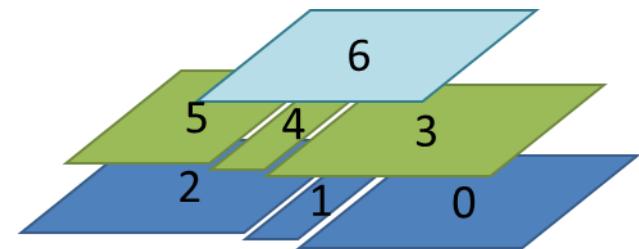
- **For faster searching,** cluster the QG by height and layer them



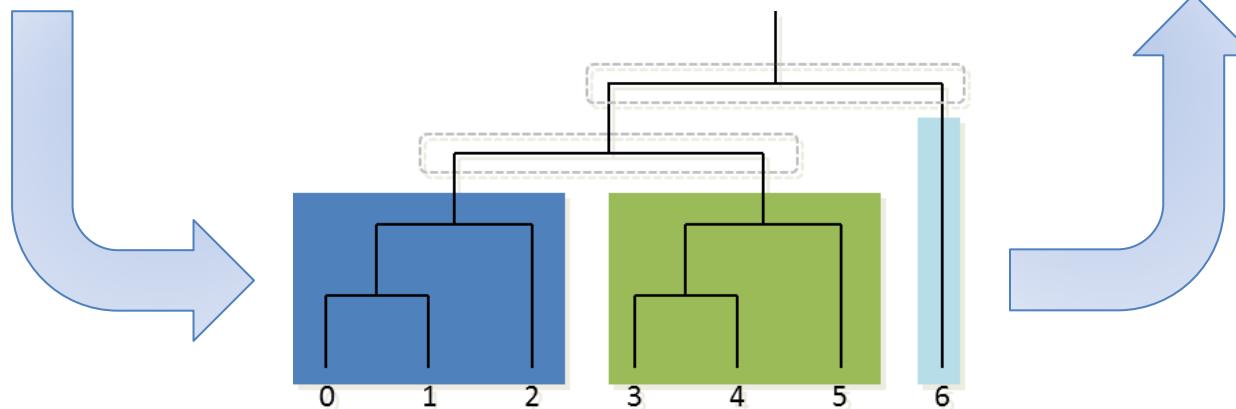
Clustering Quadtree Grids



1. Quadtree Grids for each space



3. Layered Quadtree Grids

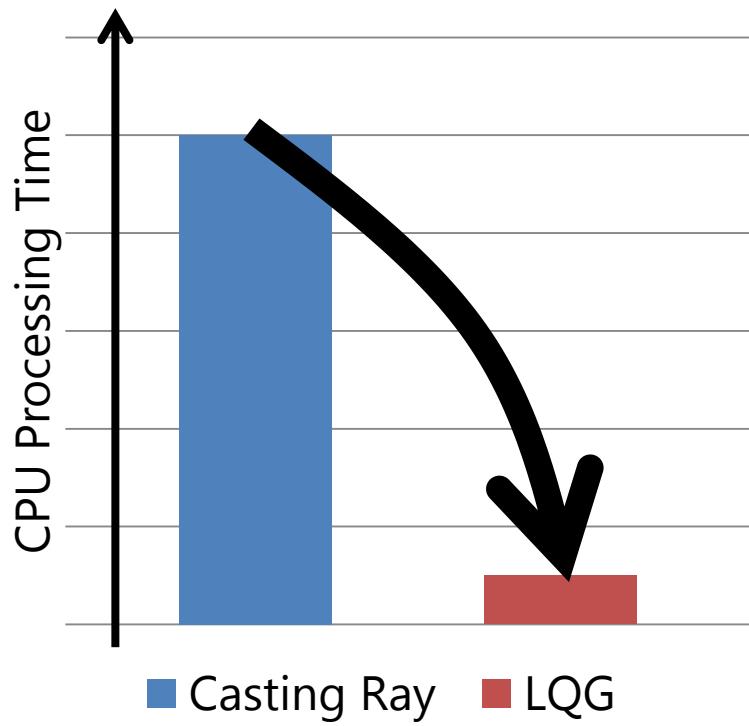


2. Cluster QG using the nearest neighbor method
for those below a certain height



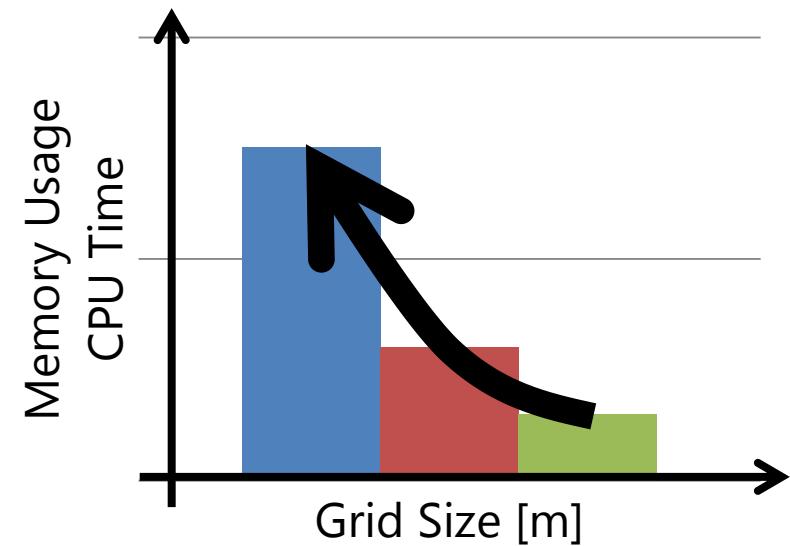
Expectations

Decreased CPU costs
using LQG rough calculation



Concerns

Increasing memory & CPU costs
for smaller grids



The smaller the grid size,
the more grids per layer



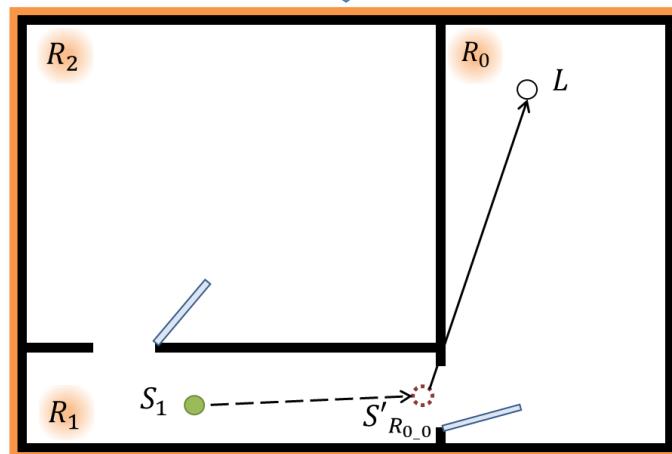
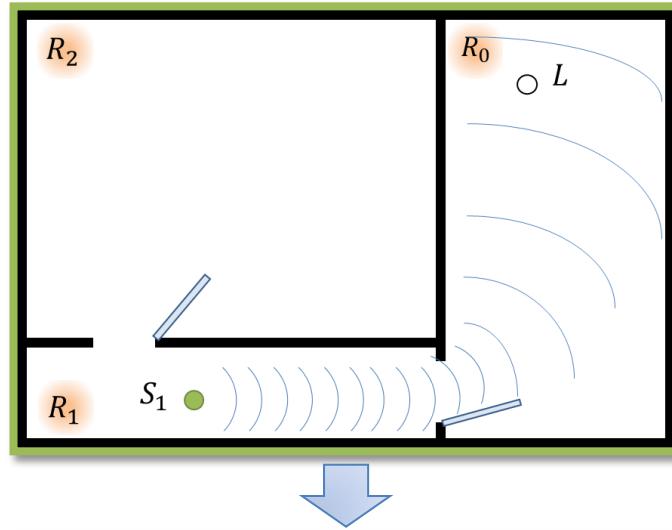
Contents

1. Introduction
2. Determining Sound Space Using LQG
3. Designing Propagation Using Rearrangement Method
4. Evaluation
5. Conclusion



Idea

How to represent propagation in more detail



Q. How do you hear **sound from another room?**

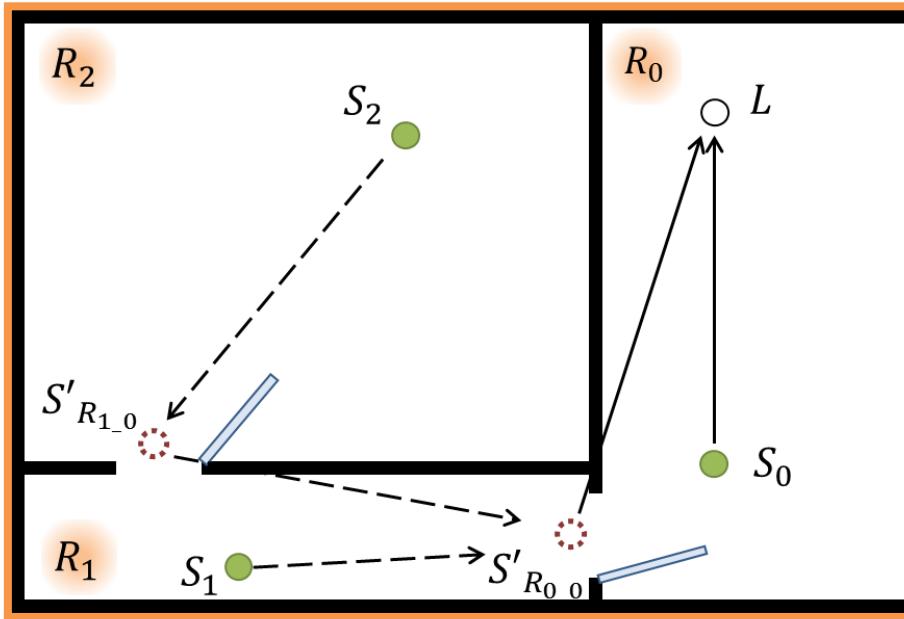
A. You hear the sound **through the connecting spaces.**
It is diffracted from other spaces given their features.

Simplification

Rearrange source in the connecting part.
Send it to the configured buses to account for the **connecting spaces**.

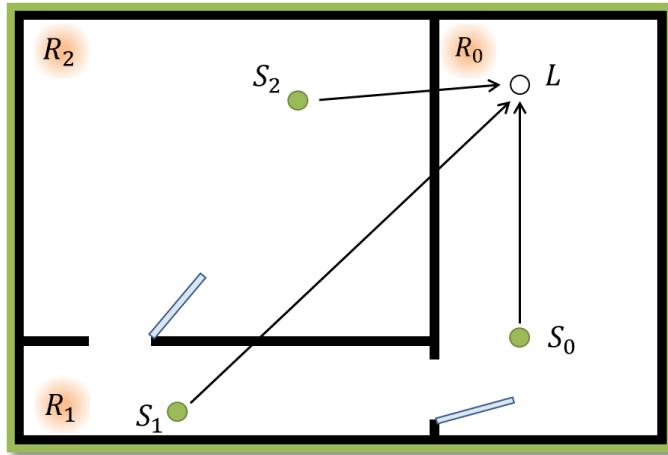


Sound level of rearrangement sources

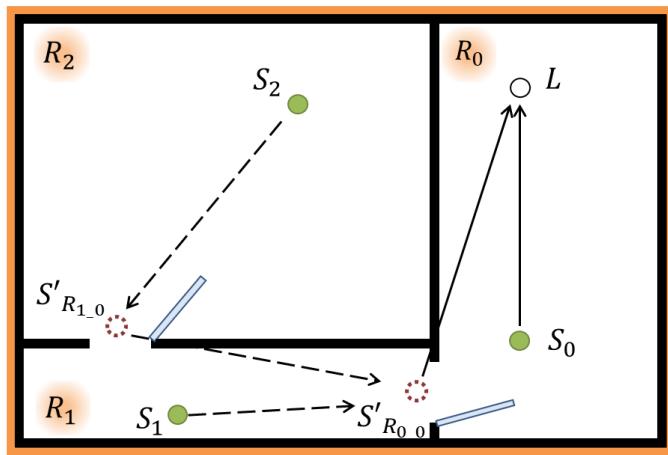
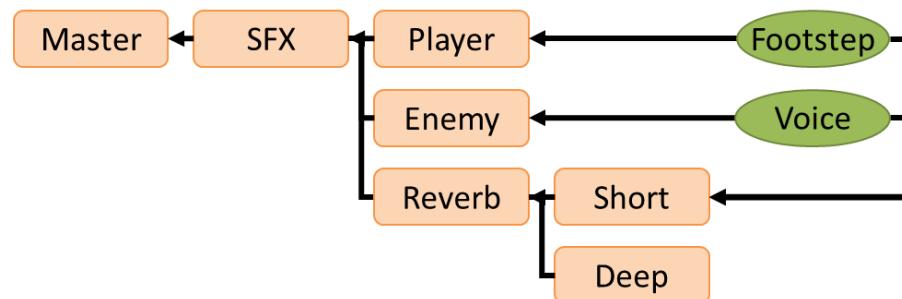


Original Position	Sound Level	Rearranged Position
S_0	L_{S_0}	-
S_1	$L_{S'_{R0_0}} = L_{S_1} - D(S_1, S'_{R0_0})$	S'_{R0_0}
S_2	$L_{S'_{R0_0}} = L_{S_2} - D(S_2, S'_{R1_0}) - D(S'_{R1_0}, S'_{R0_0})$	S'_{R0_0}

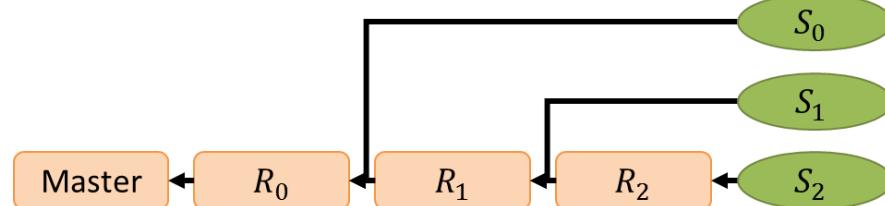
Audio Bus Structure and Effects (1/2)



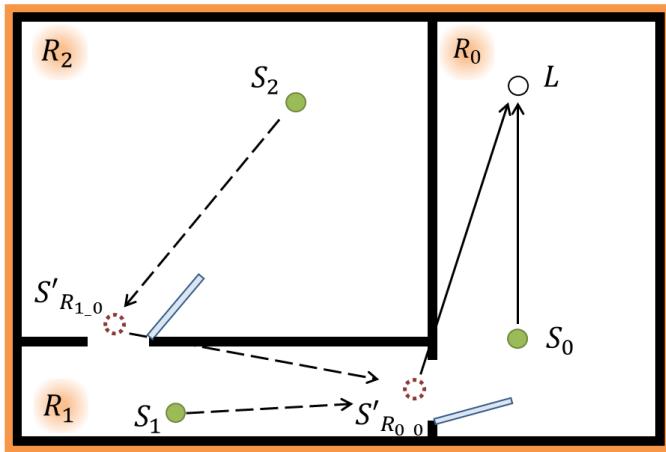
Conventional bus structure
taking into account **source types**



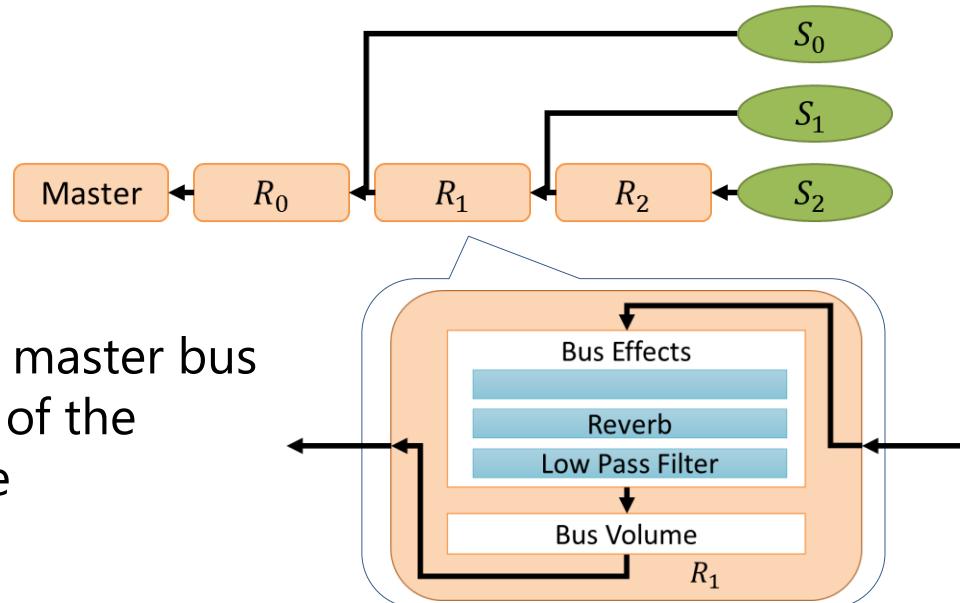
Proposed bus structure
taking into account **space connections**



Audio Bus Structure and Effects (2/2)



Create audio buses **for each space**, connect the buses depending on the **space connections**.

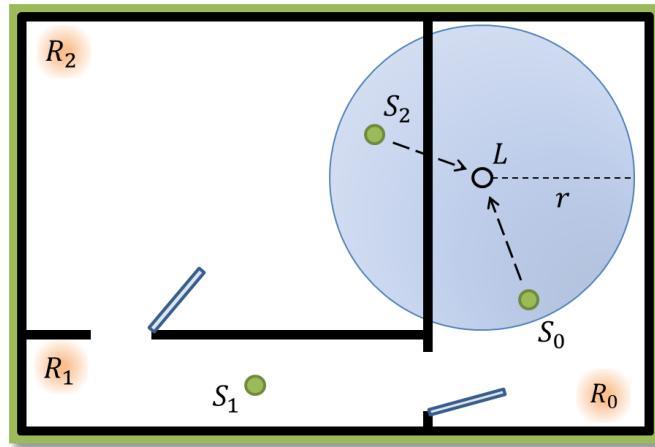


Each source goes through the master bus after being given the features of the intermediate path through the **intermediate bus**.

Instead of individually processing sound sources, process them all at once through buses.

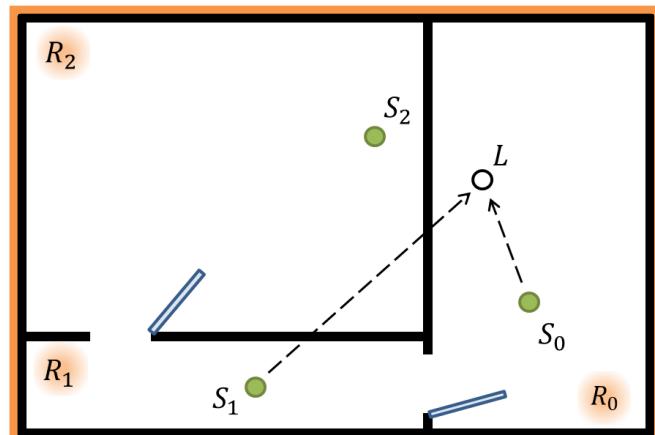


Determining Obstructions



Conventional Certain Range Method

Casting ray to the listener from sources
with distance less than r from the listener



Proposed Rearrangement Method

Casting ray to the listener from sources
in the same or adjacent space as the listener
Omit the process for ineffective sources



Expectations

- **Representation**
 - Sound carrying
 - Intermediate propagation path
- **Performance**
 - Processing sources collectively on the bus reduces processing cost
 - Reduces unimportant rays
- **Customizability**
 - Controls accuracy of representation & performance

Concerns

- **Representation**
 - Are these full enough?
- **Performance**
 - Are these light enough?



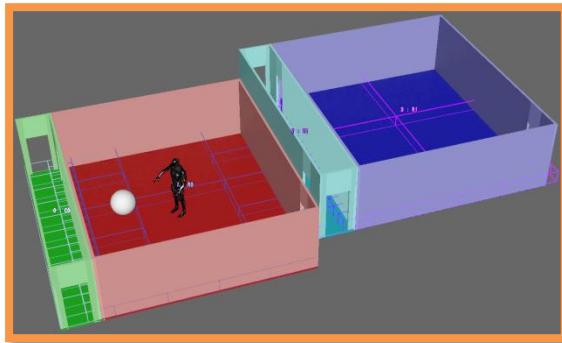
Contents

1. Introduction
2. Determining Sound Space Using LQG
3. Designing Propagation Using Rearrangement Method
4. Evaluation
5. Conclusion



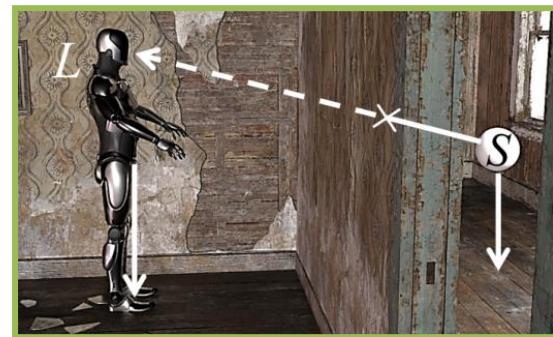
Overview

Exp 1: Determine Spaces



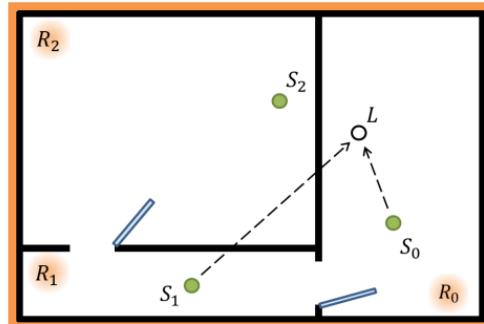
LQG method

vs



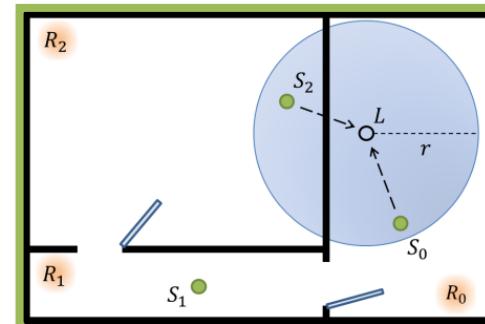
Casting ray method

Exp 2: Determine Obstacles



Rearrangement method

vs



Certain range method



Experimental Procedures



RE ENGINE

Capcom's in-house game engine

Simulated Building

4 floors (3F+B) totaling 20 rooms,
about 1,800 square meters

1. Create LQG with grid size S
2. Randomly arrange a listener and 100 sound sources
3. Determine the spaces of listener and sources using
 - i. **LQG method**
 - ii. **Casting ray method**
4. Determine the obstacles between listener and sources using
 - i. **Rearrangement method**
 - ii. **Certain range method**

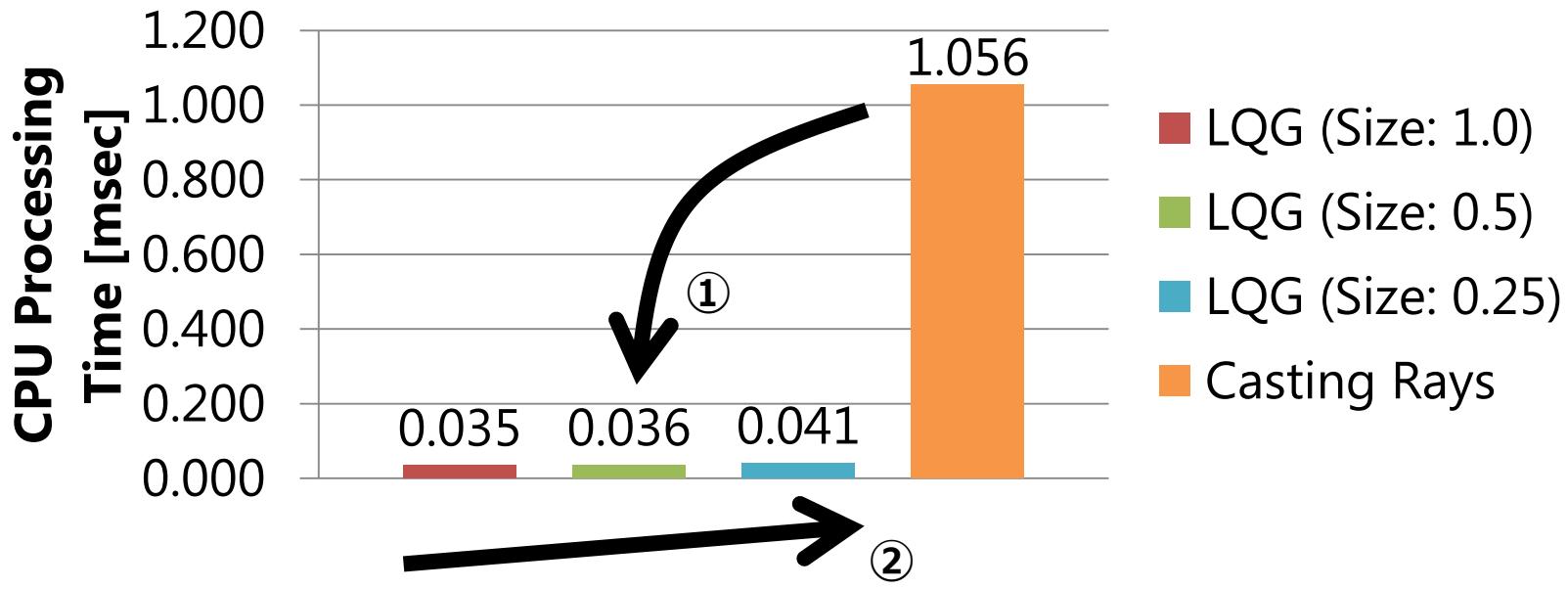
} **Exp 1:** Measure CPU processing time
& LQG Memory usage

} **Exp 2:** Measure CPU processing time



Results of Experiment 1

Determining spaces of 100 sources and a listener



① **LQG** determined the space **faster** by **1 msec** or more

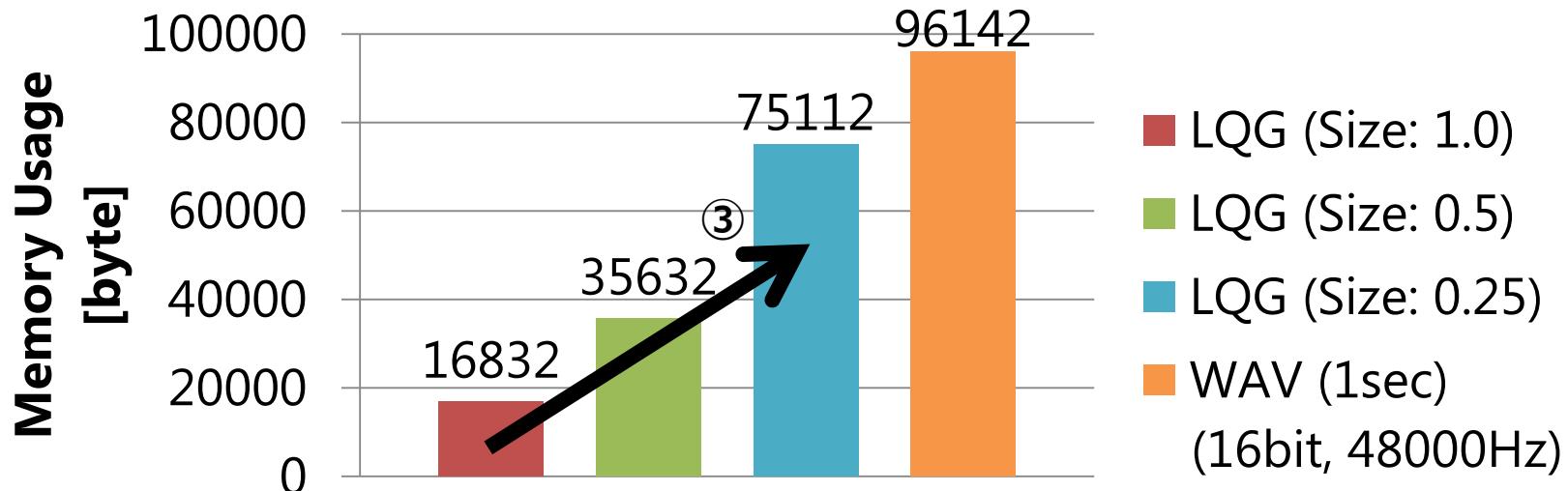
② **Processing time slightly increased** inverse to **grid size**

Sufficiently Fast Determination



Results of Experiment 1

Determining spaces of 100 sources and a listener



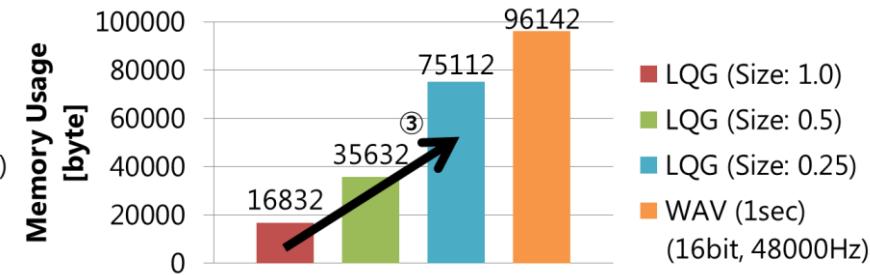
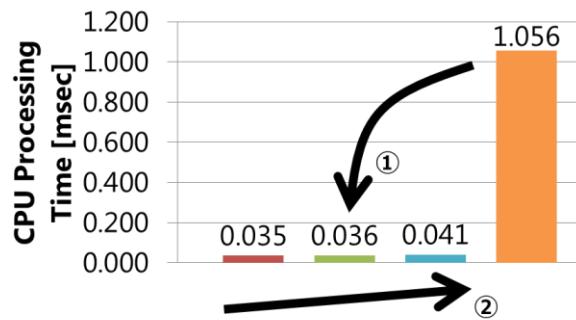
③ **Memory Usage increased** inverse to **grid size**

Sufficiently Small Memory Usage



Results of Experiment 1

Determining spaces of 100 sources and a listener



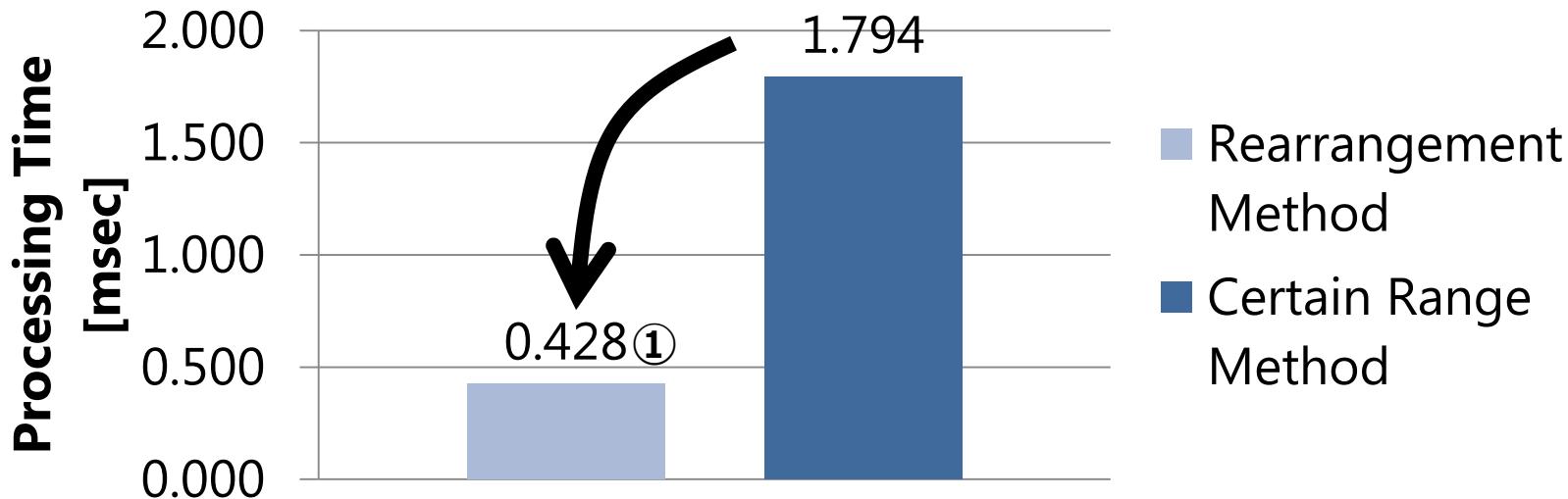
- ① **LQG** determined the space **faster** by **1 msec** or more
- ② **Processing time slightly increased** inverse to **grid size**
- ③ **Memory usage increased** inverse to **grid size**

Sufficiently Fast Determination of Space
Sufficiently Small Memory Usage



Results of Experiment 2

Determining obstacles between 100 sources and a listener



- ① **Rearrangement** determined the obstacles **faster** by **1.3 msec** or more

**Rearrangement method reduces the processing cost
by processing only the important rays**



Contents

1. Introduction
2. Determining Sound Space Using LQG
3. Designing propagation Using Rearrangement Method
4. Evaluation
5. Conclusion



Summary and Future Work

LQG

- Improvement of performance
 - Sufficiently **faster** space **determination**
 - Sufficiently **small memory** usage

Rearrangement method

- Improvement of representation
 - **Sound carrying** from other space
 - Feature of **intermediate propagation path**
- Improvement of performance
 - Decreasing of process for unimportant sources

Future work

- Improvement of representing propagation
 - Processing **sources in the same space** as the listener with **richer quality** by using more data; ex) geometry shape
 - **Preliminary calculations** of early reflections according to the position of the listener and sources

If you have any questions and so on, please contact us.