3D Audio in Games

Michele Pirovano

Contents

- Why 3D Audio in games… now?
- State of the Art of Audio in Games
  - Geometry
  - Propagation Modeling
  - Rendering/Auralization
- Improving Sound in Games: realtime 3D Audio
  - Visibility-based Beam Tracing
  - Binaural Rendering through Headphones

Why 3D Audio in Games?

- 3D audio (like 3D video) can enhance the gaming experience
  - Greater immersion
    - *«I can hear the river flowing behind me»*
  - Heightened perception
    - *«I can sense enemies coming from that direction»*
  - New interaction modalities
    - *«I can move around this maze with my eyes closed»*

[Tozer 2010]

- Many videogame genres can benefit
  - First Person Shooters (localization of enemies, immersion)
    - Example: Half Life
  - First Person Adventure (horror) games (atmosphere)
    - Example: Amnesia
  - Massive Multiplayer Online games (localization)
    - Example: Guild Wars
  - Realistic racing games (immersion)
    - Example: Formula 1
  - «Serious» games (virtual reality training)

Why 3D Audio… NOW?

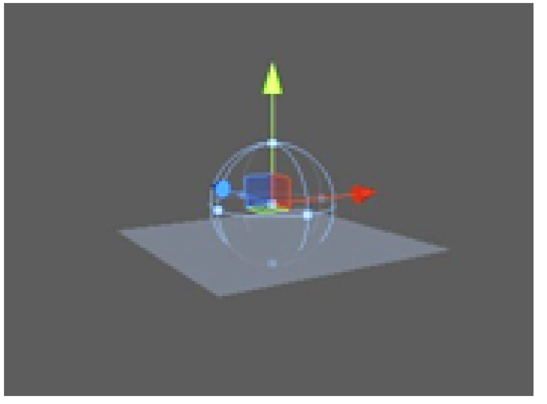
- Traditionally, little focus on audio in games due to the race for graphics **[Kapralos 2008a]**
- The focus in the gaming industry is now moving away from graphics
  - Graphics have reached their pinnacle
  - More resources are available
  - Something else is needed to make a game stand out

- Keep in mind the requirements for any new game feature:
  - Real-time constraints
  - Anything introduced in a game needs to have an impact on the game experience **[Scheel 2006]**
  - If not performed correctly, a feature may make the game experience worse
    - For 3D video: lowered performance and lower enjoyment **[Jaffe 2013]**

- *Side note: same issues as with Game AI!*

State of the Art: Geometry

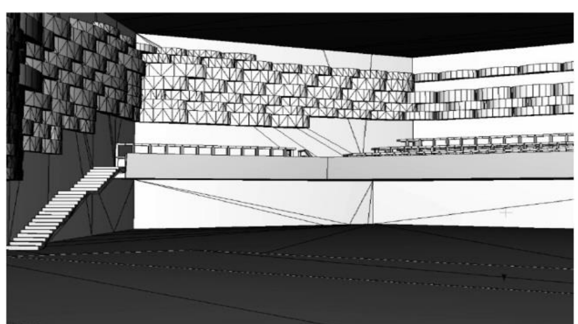
- Placement of point sources and receivers (omnidirectional)



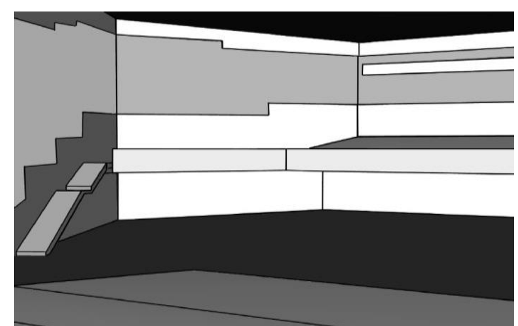
- Material properties are largely ignored
- Geometry content is already available in the form of 3D meshes
  - For sound, less detail is needed (see next slide)

State of the Art: Geometry (cont.)

- Sound wavelengths are 6 orders of magnitude longer than light wavelengths
- Due to diffraction, sound waves can bend around small objects
  - Voice frequency is between approximately 300 Hz and 3400 Hz
  - Speed of sound is 343 m/s
  - Voice wavelengths are approximately between 10 cm and 1 m
- Simplification can be performed on the 3D meshes
  - This is not so easy
- Better yet, *low poly* meshes should be created by the graphics artists beforehand
  - This is often aready done for objects that may appear far away (Level-Of-Detail-Zero)
  - Also already done for the geometry used in collision checks



Visual Geometry



Acoustic Geometry

[Anish 2009]

State of the Art: Propagation

- Note that most games:
  - are highly dynamic (moving sources, moving receivers, even moving scene geometry)
  - require real-time computation
  - on the other hand, do not need high accuracy

- Typical solution: simple attenuation on distance, no reflection nor diffraction: ignoring Room Impulse Responses (RIRs) **[Cowan 2008]**

State of the Art: Propagation

- Actual propagation can be (and has been) simulated in real-time
  - Beam tracing **[Funkhouser 1998, Anish 2009]**
    - No moving of sources nor environment
    - Receiver can be moved in real-time
      - Interactive path walking (fast look-up)
    - Supports specular reflections (extends to diffraction)

- Still not widely used in games, as sources need to move in most games!
  - Think of a First Person Shooter, where enemies move around the map

State of the Art: Rendering/Auralization

- Rendering is simple, usually in mono or, at best, in stereo
  - Amplitude panning (simulates Interaural Level Differences)
- Conveying sound
  - Headphones
  - Loudspeakers Pair
  - Sorround-sound systems
    - Limitations:
      - Higher cost and difficult setup -> not for every home
      - No full 3D (only the azimuth is concerned)
      - Works only in the "sweet spot" -> no multiple users

} Typical home configuration

[Tozer 2010, Kapralos 2008a]

Example: Unity

- Common middleware for videogame creation
- Uses FMOD audio library
- Geometry
  - Placement of AudioSources and AudioListeners
- Propagation Modeling
  - Simple distance-based relation
  - Simulation of attenuation (rolloff) and directionality
  - Filters can be applied
  - Ignores diffraction and reflection (no tracing at all)
- Rendering
  - Amplitude panning on loudspeaker pair or headphones

Improving the State of the Art

- Geometry is good enough
  - Use of existing (simplified) geometric scene
- Propagation modeling can be improved
  - Wave methods? Geometric methods? (See next slides)
- Rendering can be improved
  - HRTF modeling for localization (See next slides)
  - Binaural Rendering through headphones

Propagation Improvements

- Wave-based modeling
  - Solving the wave equation is overkill for a game
    - Too much accuracy, too costly computation
- Geometric methods
  - Ray tracing -> aliasing, inefficient
  - Virtual sources -> few games are in a simple rectangular room
  - Beam tracing
  - **Visibility-based beam tracing**
    - Retains benefits of beam tracing
    - If source changes position (but is still static), we can recompute the beam tree efficiently
    - Able to move source and receivers: works well with game dynamics

[Antonacci 2008, Markovic 2010]

Rendering Improvements: HRTF

- A Head Related Transfer Function models the response of an ear to the sound signal from a point in space
  - Depends on the shape of the upper torso, shoulders, head and pinna -> different for every person
  - Measured with «dummy» heads
    - Hard to measure at home for the player
- The HRTF is dependent on frequency **ω** and direction **θ**
  - After 1 meter, we can ignore distance (far-field HRTF)
- Generalized HRTFs can be used for games
  - Obtained by averaging HRTFs
  - May present problems such as localization errors, localization blur, externalization errors, front-back confusion
  - Head tracking can be useful to provide further localization cues

[Kapralos 2008a, Tsakostas 2007]

Chosen Algorithm

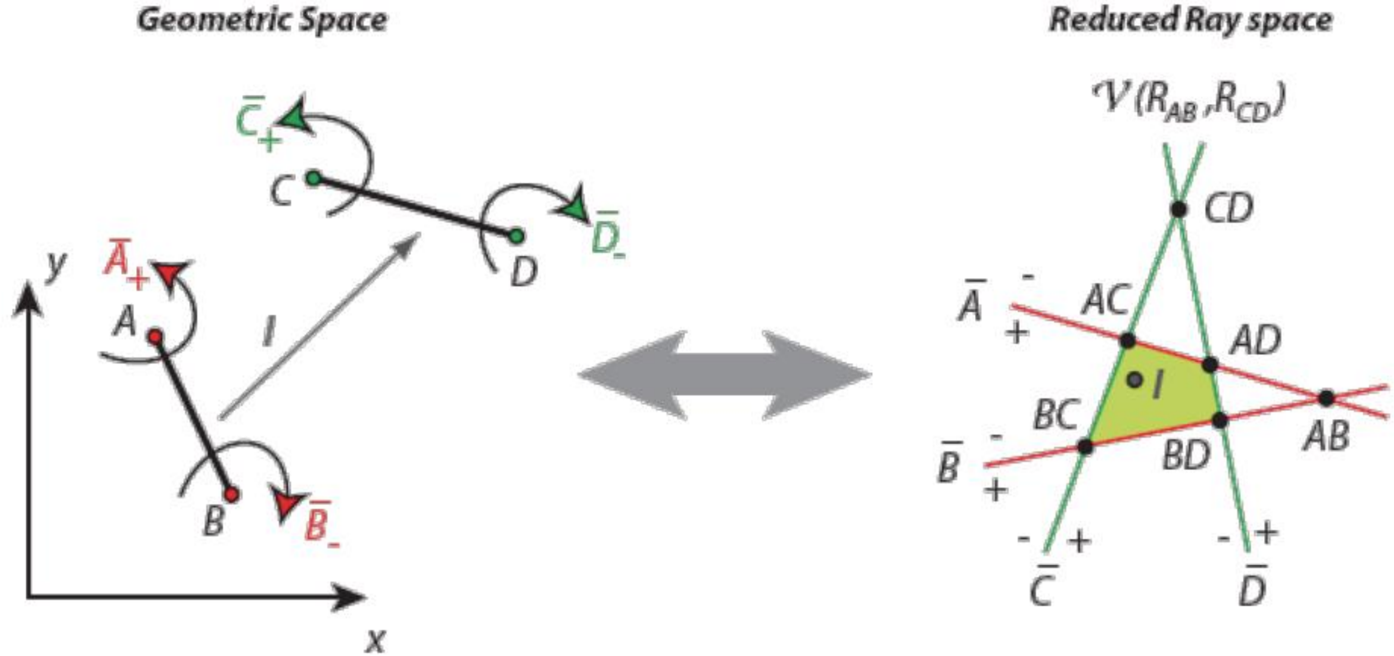
1. Start from the simplified geometry of 3D meshes
2. Build the global visibility graph

Pre-computed (steps 1–2)

3. Propagate throught visibility-based beam tracing
   – obtain a beam tree for each (image or real) source
4. Trace paths between sources and receiver
   – obtain a collection of directional beams to the receiver
5. Convolute with HRTFs for localization
6. Binaural rendering through headphones

Visibility-based beam tracing (1)

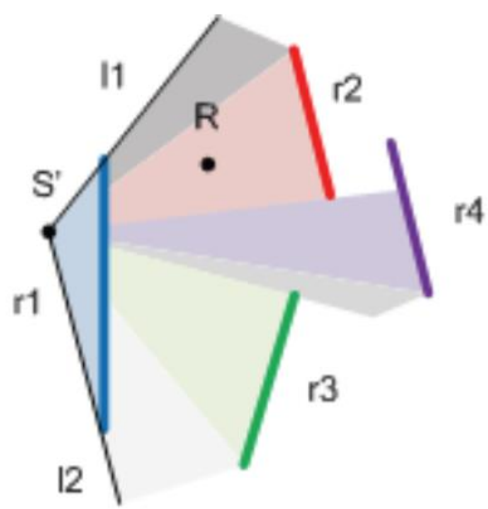
- Map reflectors to the Ray Space
- Obtain global visibility in the form of a Visibilty Diagram *V(R$_i$)* for each reflector *R$_i$*

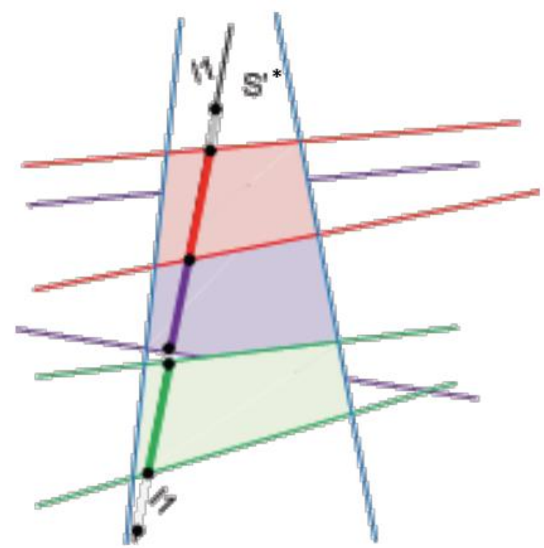


Visibility-based beam tracing (2)

Place sources and compute the beams for each source and its images

- Done by scanning visibility diagrams -> fast look-up
- Chosen recursion level for speed/accuracy trade-off
- Builds beam tree of source visibility

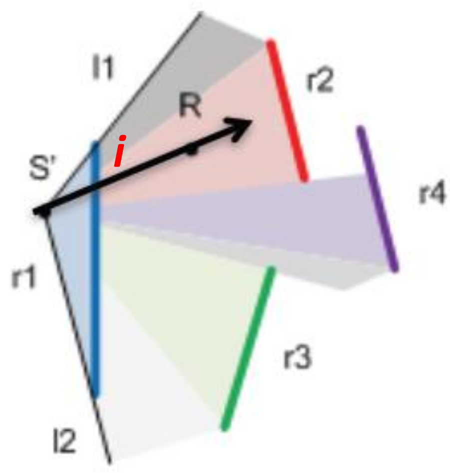




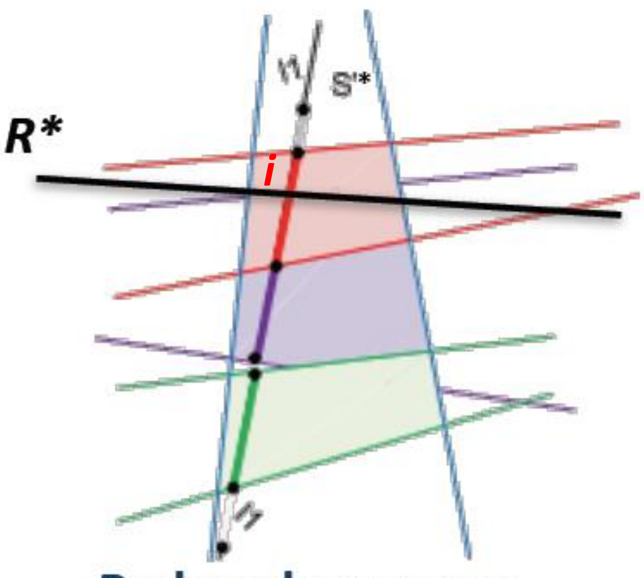
Visibility-based beam tracing (3)

Place the receiver (player) and trace the path of the ray *i* going from each (image) source to the receiver

- Path tracing through ‘dual’ line intersection -> again a fast look-up



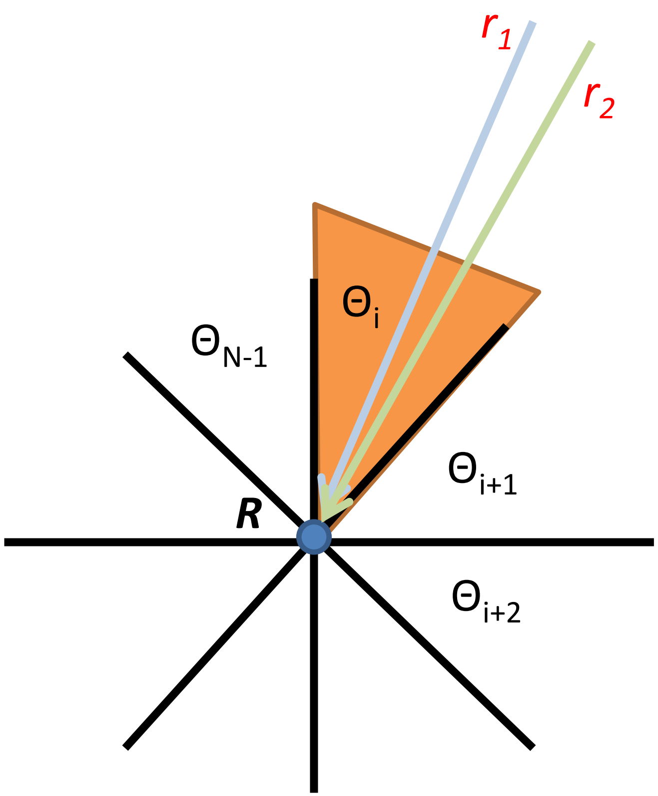
Geometric space



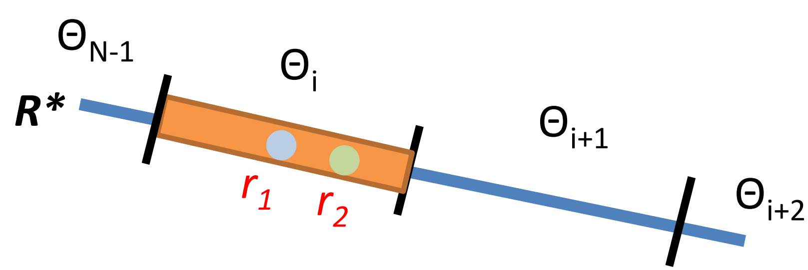
Reduced ray space

Visibility-based beam tracing (4)

- Collect incoming rays into N directional beams
  - $b(t,\Theta_i)$ will represent the accumulated signal in a beam



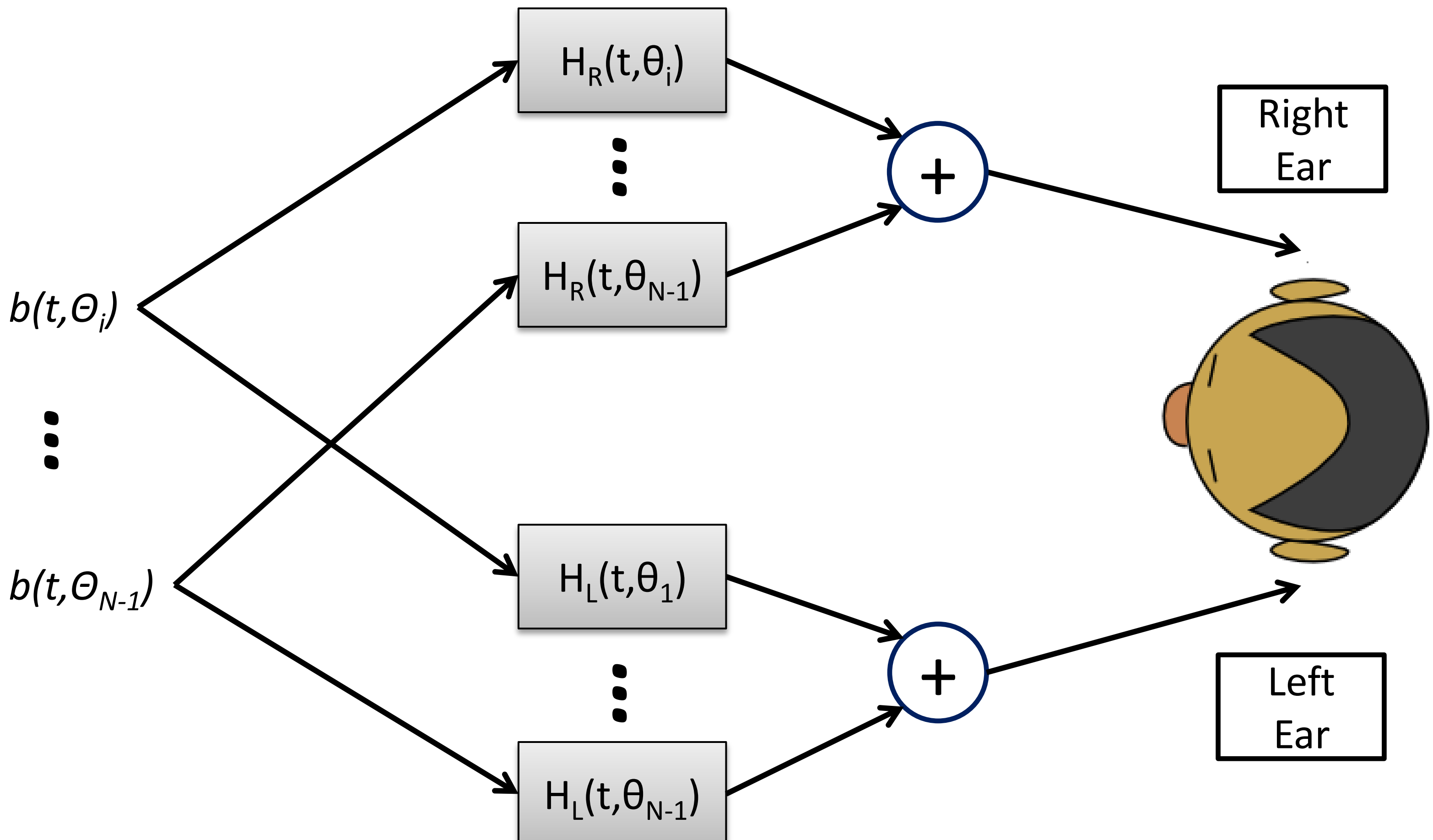
Geometric Space



Reduced Ray Space

HRTF convolution

- Convolute the signals through the HRTF for each ear and angle sector to get the final signals



HRTF convolution

- Convolution requires a lot of computation, but…
  - We can lower the dimensionality of the (usually large) HRTF data sets
    - **[Kapralos 2007, 2008b]**
  - We can speed-up the computation using the GPU
    - **[Hamidi 2009, Cowan 2008]**

Conveying Sound

- Once the final sound signal is ready, binaural rendering can be performed through
  - Headphones
    - Sending the correct signal to each ear enables localization
  - Why not a Loudspeakers Pair?
    - Need for crosstalk cancelation **[Kapralos 2008a]**
      - Can be done in real-time, but it works only in a small sweet spot!
- There is no need for a more complex setup (such as surround-sound systems), since we perform fully-computed auralization **[Kleiner 1993]**

References

- *Tozer, J. - 2010 – An exploration the effectiveness of 3D Sound on video games*
- *Jaffe, S. – 2013 - Effects of 3D Audio and Video in Video Games*
- *Anish, C - 2009 - An Introduction to Sound Rendering*
- *Antonacci, F. et al. – 2008 - Fast tracing of acoustic beams and paths through visibility lookup*
- *Kapralos, B. et al. – 2007 - Dimensionality Reduction Techniques to HRTFs for Interactive Virtual Environments*
- *Kapralos, B. et al. - 2008a - Virtual Audio Systems*
- *Kapralos, B. et al. – 2008b - Dimensionality Reduced HRTFs: A Comparative Study*
- *Hamidi, F. and Kapralos, B – 2009 - A Review of Spatial Sound for Virtual Environments and Games with Graphics Processing Units*
- *Cowan, B. and Kapralos, B. - 2008 - Spatial Sound for Video Games and Virtual Environments Utilizing Real-Time GPU-Based Convolution*
- *Kleiner, M. et al. - 1993 - Auralization: An Overview*
- *Markovic, C. et al. – 2010 - Visibility-Based Beam Tracing for Soundfield Rendering*
- *Funkhouser et al. - 1998 - A beam tracing approach to acoustic modeling for interactive virtual environments*
- *Tsakostas, C., Floros, A., & Deliyiannis, Y. - 2007 - Binaural Rendering for Enhanced 3D Audio Perception.*
- *Schell – 2006 – The Art of Game Design: A Book of Lenses*

Thank You!