Virtual Reality Technology and Programming

TNM053: Lecture 11: Audio in VR

Audio in VR
- Audio is very important in VR
- High Quality audio provides:
  - Increased realism
  - Strong immersive sense
  - Strong positional cues
  - Extra information about the environment
    - The shape of the world
  - What does ‘High Quality’ mean?

VR sound environment
- VR equipment is a really bad place to be trying to create sound.
- CAVE
  - Stand in a glass box
  - Pretend you can’t hear the echoes.
  - Also hard to place speakers

VR sound environment (2)
- Semi-immersive VR theatre
  - Sit in a big cylinder
  - Reflects sound in a very strange way.

Workbench
- Not so bad but still...
- Big flat screen 1 metre in front of you
- Sound coming from surround speakers
- Creates echoes inappropriate for scene

Audio equipment in VR
- So, most of VR audio is based on very high quality headphones
- Use head tracking to get position and orientation
- Generate sounds as if from sources
- Play to user
- Problem solved!
- ...sadly, not.
A world of sound

- You are surrounded by sound all the time – Silence is unheard of!
- The environment affects (shapes?) the sound you hear
  - Size
  - Shape
  - Materials

Anechoic chamber

- No echoes
- Quiet!
  - So quiet you can feel it!
- No superposition
  - Voices sound *very* different
  - Quieter, different frequencies, no echoes
- All effects you are used to hearing
  - Their absence is very noticeable

Consider the lecture theatre

- Designed to channel sound from the speaker to the audience
  - Shaped to funnel sound to the back

What do you hear?

- Amplitude

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>( t_0 )</th>
<th>( t_1 )</th>
<th>( t_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Sound</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First Reflection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Echo...echo...echo...

- Sound impulses produce echoes in all normal environments
- So does speech
- Time integration is shorter than the decay time
  - You hear some of the echoes
  - Probably not shorter than \( t_1 - t_0 \)
  - You don't hear all of them!
- Normally you don't mind.
  - You ignore the superposition
  - You ignore the frequency changes
  - (Could be a cause of stammering)

Rendering sound
- ‘Auralization’

- To generate correct echoes must model sound behaviour in the space
- Rooms are complex
- Filled with different materials
  - Reflective
  - Absorbant
  - Frequency filtering
- Just like rendering light
Illumination modelling

- Many methods
  - Ray tracing
  - Radiosity modelling
- More complex methods
  - Image based lighting
  - Others

Ray tracing

Audio ray tracing

- Doesn't work
- Sound is slow
- Sound really doesn't travel in straight lines
  - neither does light... but we don't notice (much)
- Different frequencies don't behave the same
  - neither does light... but we don't notice (much)
- All objects in scene filter sound affecting:
  - reflection
  - diffraction
  - frequency

Radiosity mapping

- Use patches
  - model the incident light ‘energy’
  - compute how those patches illuminate other patches
- Much more efficient but based on similar approach
- Light travels in straight lines
- ...or appears to most of the time

Diffraction problem - Light

Diffraction problem - Sound
Reflection problem - Light

- Incident ray -> reflected ray + ambient light

Reflection for real light

- No such thing as ambient light
- Real surfaces are not planes
  - They have surface properties
  - Skin has thickness and a fat layer

Reflection problem - Sound

- For sound *all* materials have very complex surface properties:
  - ‘skin’ and ‘fat’
- Surfaces resonate
- Surfaces diffract
- All sound reflection is hard to calculate
  - Not yet a real-time problem

Motion makes it worse!

- Sound is slow
  - ~300m/s (c.f. light: ~300 000 000m/s)
- 100kph = ~27m/s
- Doppler effect changes frequencies from a moving sound source
  - ~10-30% when a car passes you
- 10KPH = ~2%

Putting sound in VR environments

- What sound?
- ‘Ambient’ sounds
  - ‘Surround’ sound
  - Often use recorded sounds
- Positional sounds
  - Designed to give a strong sense of something happening in a particular place
  - Also often provided by using recorded sounds

Positional sound

- Using sound to create the sense of active things in the environment
  - Enhances presence
  - Enhances immersion
- Need to deal with many components
  - Reflections (echoes)
  - Diffraction effects
Consider VisClim
- Scene in Linköping's Storatorget
- Surrounding environment
  - Vehicles? - Several roads nearby
  - People? - Many people in the square
- Weather noise effects
  - Rainfall
  - Snowfall (no sound but damping effect)

VisClim

Air traffic control
- Simulation so
  - No 'ambient' sound required
  - No aircraft noises
  - No realism wanted?
- Positional warnings?
  - Designed to draw the users attention to the location of a problem
  - Which may be out of the field of view

Air traffic control

Creating positional sound
- Amplitude
  - stereo (or more)
- Synchronisation
  - Audio delays
- Frequency
  - HRTF

Positional sound - amplitude
- Generate audio from position sources
- Calculate amplitude from distance
- Include damping factors
  - Air conditions
  - Snow
  - Directional effect of the ears
Positional sound - Synchronisation

- Ears are very precise instruments
- Very good at hearing when something happens after something else
- Use this to help define direction
  - Difference in amplitude gives only very approximate direction information

Positional sound - Synchronisation

- 30 centimetres
- =0.001 seconds!
- Human can hear = 700µS

3D positional sound

- Humans have stereo ears
  - Two sound pulse impacts
  - One difference in amplitude
  - One difference in time of arrival
- How is it that a human can resolve sound in 3D?
- Should only be possible in 2D?

Positional sound - Frequency

- Frequency responses of the ears change in different directions!
- You hear a different frequency filtering in each ear
- Use that data to work out 3D position information
- How do we mimic that in headphones?

Head-Related Transfer Function

- Define a frequency transfer function
  - for each ear
  - for every direction around the head
- HRTF varies from person to person
  - Must measure each individual user!
  - Measurements involve microphones inserted deep in the user’s ears.
  - Use movable sound source to measure response

Using HRTF’s
HRTF's

- HRTF's are 3D
- Depend on ear shape (Pinna) and resonant qualities of the head!
- Allows positional sound to be 3D
- Computationally difficult
  - Originally done in special hardware (Convolotron)
  - Now can be done in real-time using DSP

Audio rendering summary

- Rendering audio is really, really hard
- Much bigger problem than lighting
- Material properties are more complex
  - Can't fake it as easily
  - Properties are always a problem
- Many good methods exist but the problem is too computationally hard for general use at present

What can we do today?

- Constraint is real time audio rendering
- Simple (reflectionless) stereo positional sound
  - Using amplitude
  - Using synchronization
  - Using HRTF frequency filtering
- Useful for audio cues and simple environmental sounds

What can't we do today

- Full interactive audio rendering
- Still too large a computational problem
  - Reflections and diffractions too complex
    - for real situations at least
- Simple scenes do provide some scope
  - Buildings - hard materials, flat planes
  - Could handle VisClim in real time?

Voice interaction

- Voice input for control
  - Continuous?
  - Discrete?
- Voice output for information
  - Positional? - Alerts
  - Arbitrary? - Purely informational

Voice output

- Speech synthesis is quite sophisticated
- Can create a reasonably human-sounding synthetic voice
  - quite understandable
  - often not too pleasant to listen to
- Can use sampled sound and signal processing to create a real sounding voice
  - in more limited circumstances
Voice output
- Can combine voice synthesis with positional sound to create a warning
  - Use position as an ‘attractor’
  - Draw user’s attention to the location of the problem
- (Maybe?) no need to worry about complex sound rendering

Voice recognition
- Potentially very powerful interaction mode
- Can free the user
  - From keyboard
  - From 2D or 3D mouse
- Been in development for 20 years
  - with limited success

Voice recognition: The dream

Voice recognition: The reality

Continuous speech recognition
- Requires
  - Very good sound equipment
    - High quality microphone
  - High quality sound hardware on computer
  - Minimum of 500MHz processing power
    - Probably much more
  - Lots and lots of physical memory
    - database of grammar and word structures
### Continuous speech recognition

- Requires substantial amount of training of the recognition system
- Modern systems learn from use
  - Remembers the corrections you make
- Several systems on the market
- Error rate remains too high to be useful for interaction purposes.

### Discrete speech pattern recognition

- Designed to build simple interfaces
  - Like the one S.J. uses for customers checking train times over the telephone
- The command set is small
- Often the sounds are distinctive
  - Even with accents?
- Can exploit this for our purposes

### Discrete speech pattern recognition

- Create a set of command phrases
  - As small as possible
  - As distinct as possible ("on"/"off" are bad!)
- Use audio pattern matching to determine if one of the command phrases has been said.
- Have to get it's attention somehow
  - Like a button?

### Voice recognition in ATC

- Air traffic control application
  - Has discrete set of commands
  - Often want to turn features on and off
    - "enable" and "disable"!
  - Can build a command set
- Use button to get its attention
- Error rate <5% (on a good day)

### ATC command set

- **restart scenario:**
  - Say this part
  - This is sent to the application
  - Currently have about 90 commands
    - 25 of which are flight selection

<table>
<thead>
<tr>
<th>ATC command set (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>height scale 1:scalefactor 1</td>
</tr>
<tr>
<td>height scale 3:scalefactor 3</td>
</tr>
<tr>
<td>height scale 6:scalefactor 6</td>
</tr>
<tr>
<td>time factor 1:updateref 1</td>
</tr>
<tr>
<td>time factor 4:updateref 4</td>
</tr>
<tr>
<td>time factor 10:updateref 10</td>
</tr>
<tr>
<td>waypoints disable:waypoints 0</td>
</tr>
<tr>
<td>waypoints enable:waypoints 1</td>
</tr>
<tr>
<td>clip box disable:clipbox 0</td>
</tr>
<tr>
<td>clip box enable:clipbox 1</td>
</tr>
<tr>
<td>compass size big:instrumentsize 3</td>
</tr>
<tr>
<td>compass size small:instrumentsize 1</td>
</tr>
<tr>
<td>sector mode:sectormode 1</td>
</tr>
<tr>
<td>trajectory mode:sectormode 0</td>
</tr>
</tbody>
</table>
**ATC command set(2)***

- trajectory lines: trajstyle Lines
- trajectory tubes: trajstyle Tubes
- trajectory all disable trajmode 0 0; trajmode 1 0; trajmode 2 0
- trajectory all enable trajmode 0 1; trajmode 1 1; trajmode 2 1
- trajectory incoming disable trajmode 1 0
- trajectory incoming enable trajmode 1 1
- trajectory outgoing disable trajmode 2 0
- trajectory outgoing enable trajmode 2 1
- trajectory inactive disable trajmode 0 0
- trajectory inactive enable trajmode 0 1
- focus on flight S K 2 3 1: airplane focus SK231
- focus on flight S K 2 3 2: airplane focus SK232
- focus on flight S K 2 3 4: airplane focus SK234

**Summary: Voice interaction***

- **Speech synthesis works quite well**
  - Allows for easy informative feedback
- **Speech recognition is harder**
  - General case still impossible
- **Defined command set can be worked with**
  - Recognition rates can be very high
  - Still not perfect