Interactive 3D audio rendering systems

Nicolas Tsingos

Sound Technology Research

Dolby Laboratories Inc.

DOLBY.

Interactive 3D audio rendering

- Render audible virtual sound sources
- Many applications
  - Games
  - Acoustical design
  - Virtual/Augmented reality
- Essential for immersion
- Strong real-time constraints

© Kunstmuseum Bonn

©Eden Games

Challenges of 3D Audio Rendering

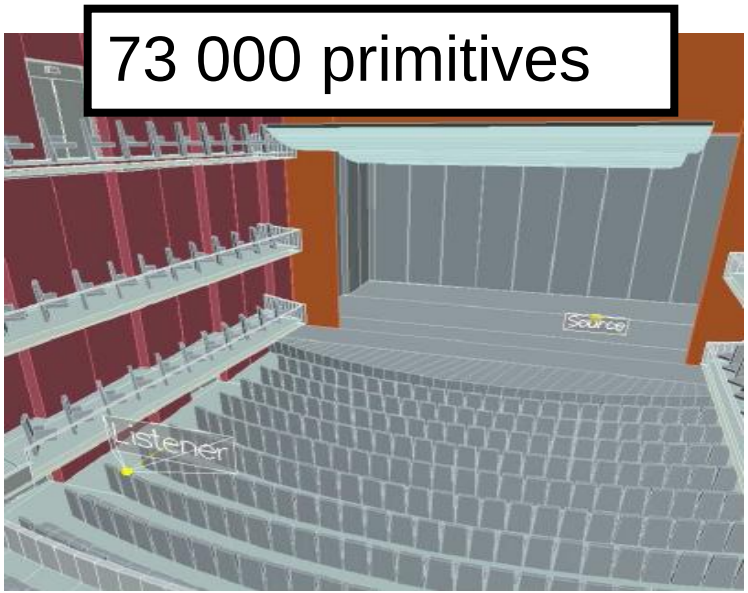
- Process more sources in real-time
- Process highly dynamic environments
- Achieve more realistic interactive effects
  - Reverberation
  - Occlusion
- Render convincing spatial audio

© Kunstmuseum Bonn

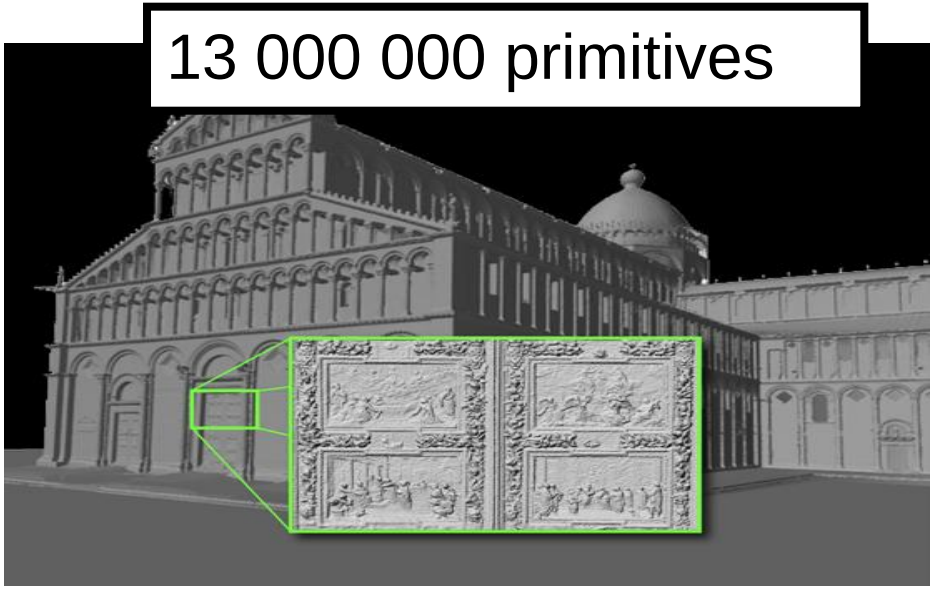
©Eden Games

Challenges of acoustical simulations

Processing complex geometry…



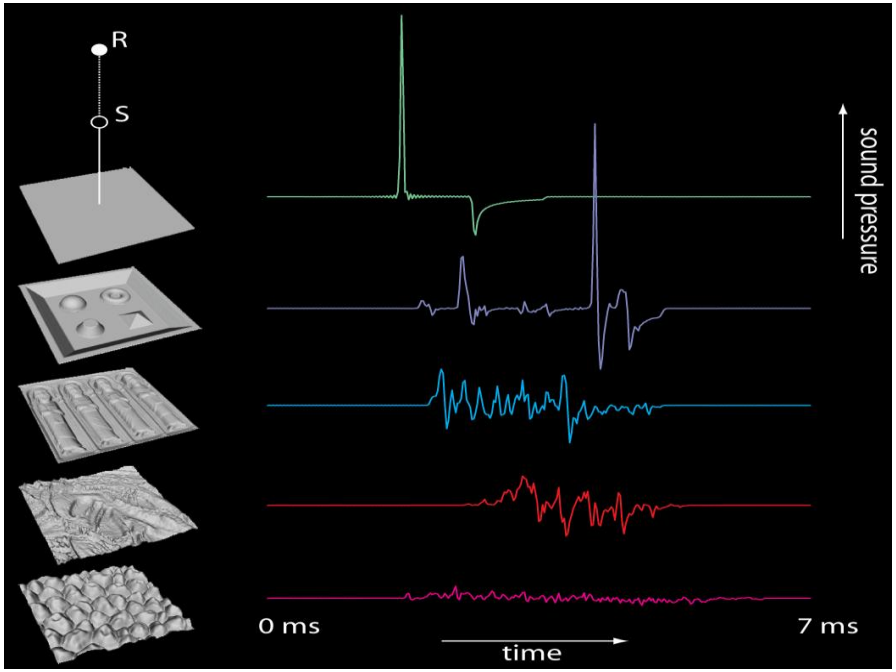
©Siltanen/HUT 2005



©Visual Computing Lab ISTI/CNR

Challenges of acoustical simulations

…requires more accurate wave-based models



Why GPU computing ?

Modern graphics cards (GPUs) are programmable

Supercomputing performance…

Hundreds of processing cores : TeraFLOP performance

Huge memory bandwidth : 100+ Gb/s.

…if your problem maps well to their architecture

it must be highly parallel

Available everywhere

Product line from notebooks to supercomputers

Outline

- Brief overview of GPUs and GPU programming
- Audio processing using graphics API
- Massive reverberation processing with CUDA
- Other applications of GPUs for audio rendering
- GPU-based audio scattering
- Conclusions

Short overview of GPUs

- Graphics Processing Unit (GPU)
  - Composed of many simple processors
  - Originally used for rendering graphics
- Designed for massively parallel processing
- Now being used in server farms and scientific computing

Short overview of GPUs

- GPU Parallel processing
  - Hundreds of simple processors
  - Hundreds of pixels on a screen

- For graphics, each processor:
  - Computes a single 3D point
  - Colors a single pixel

Short overview of GPUs

- **GPU**
  - Hundreds of concurrent threads
  - Each thread identical
  - Simple logic
  - Best for processing chunks of data
- **CPU**
  - Dozens of concurrent threads
  - Threads are independent
  - Any kind of logic
  - Best at single stream of complex logic

GPU computing approaches

Trick the graphics pipeline into doing computation

- Disguise data as textures or geometry
- Disguise algorithm as render passes
- Graphics APIs + high-level shading languages
  - openGL, Direct3D, Cg, GLSL, HLSL

Program GPU directly, no graphics-based restrictions

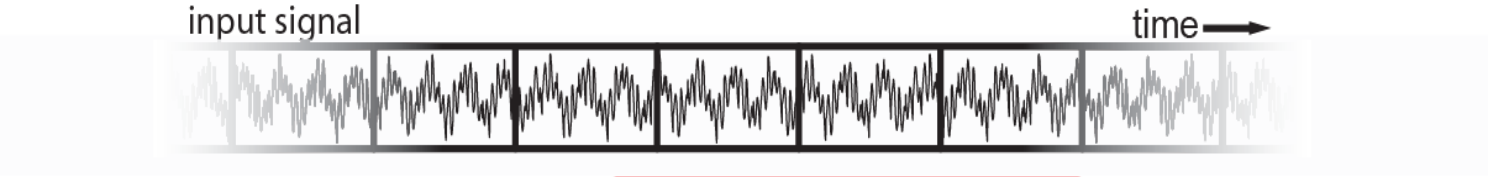
- Provide scatter, inter-thread communication, pointers
- Enable standard languages like C, Fortran, Matlab
- In 2006, NVIDIA introduces *CUDA*
- *2009: OpenCL, Direct compute*

GPU for audio processing

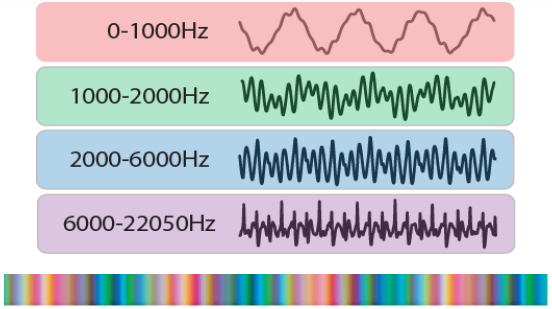
Process audio samples

3D audio processing for numerous sound sources

simple HRTF model, EQ, distance, Doppler shifting



Store audio into texture memory



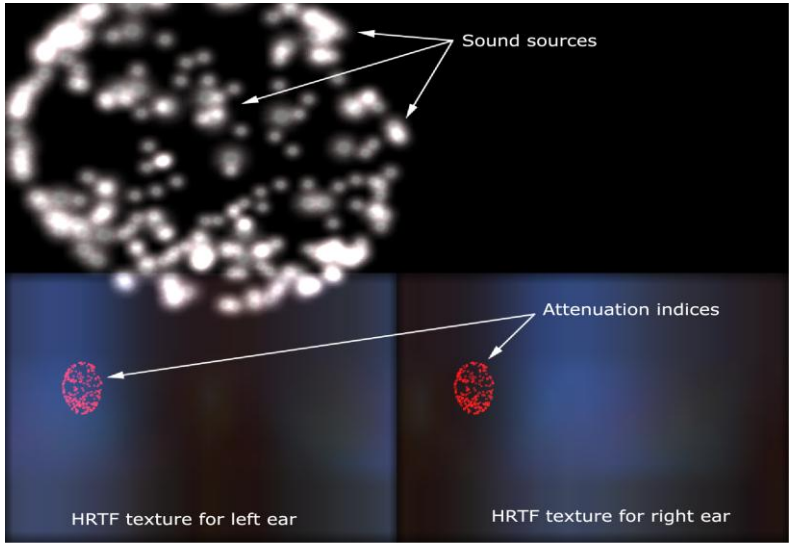
GPU for audio processing

Process audio by drawing lines or polygons

- use texturing hardware for resampling
- use dot product instructions
- use textures to store filter coefficients

Peak #sources

- 700 P4 3GHz SSE
- >2000 QuadroFX4500
- (mix rate: ~200K frames / sec.)

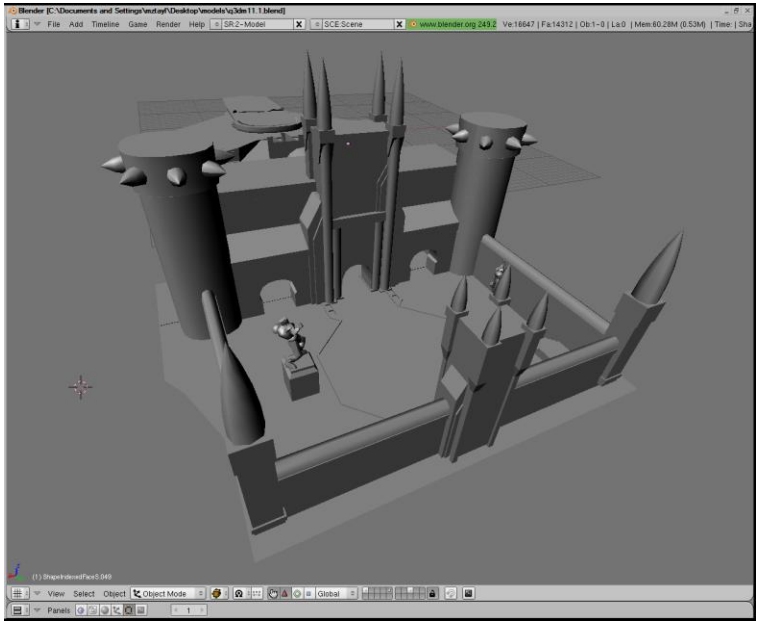


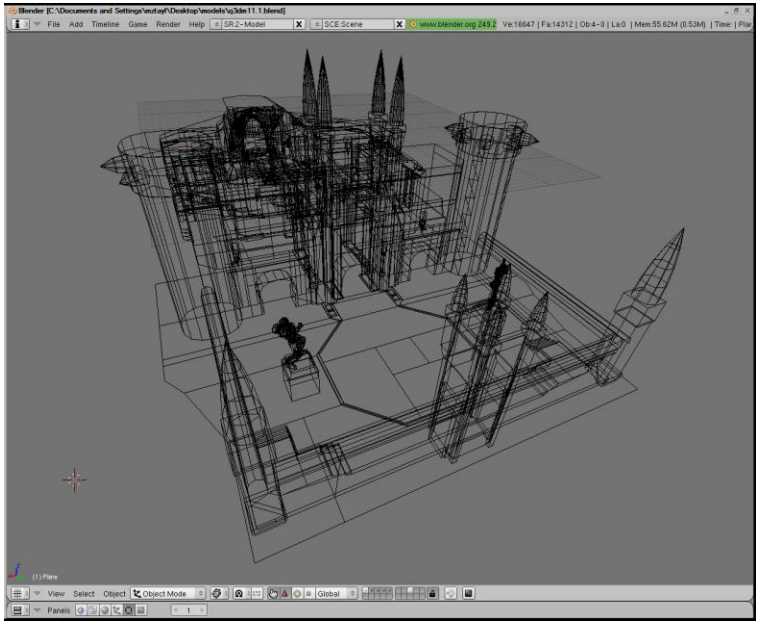
GPU for audio processing

- Reverberation and occlusion
  - Based on geometry of environment

- Need to scale to hundreds/thousands of sources
  - Games have increasingly complex physics-related sources
  - Rendering complex scenes on audio rendering servers

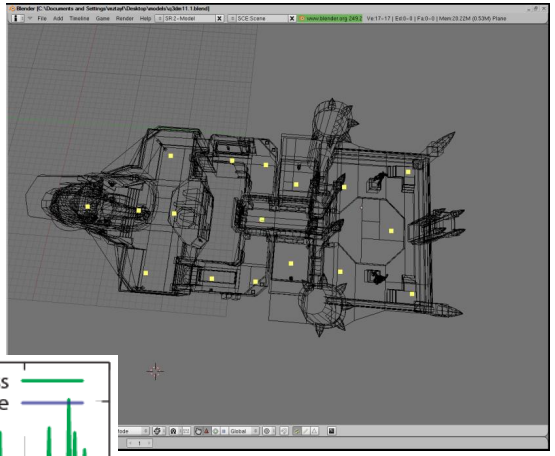
Modeling environmental effects

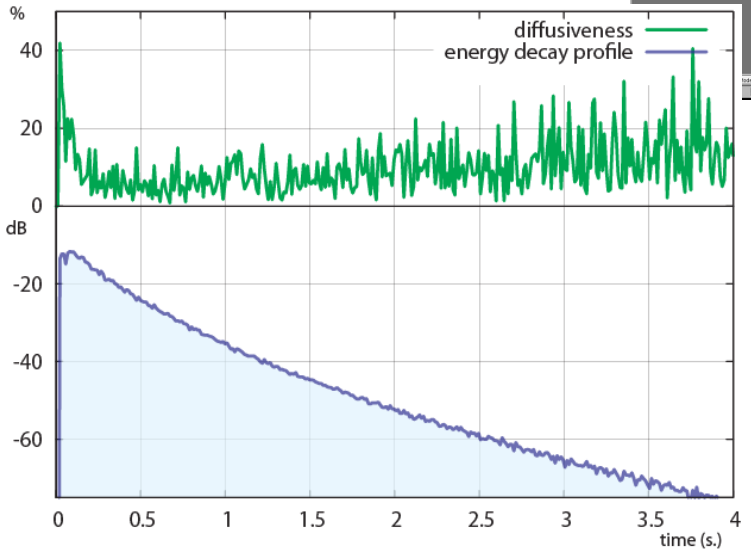




Modeling environmental effects

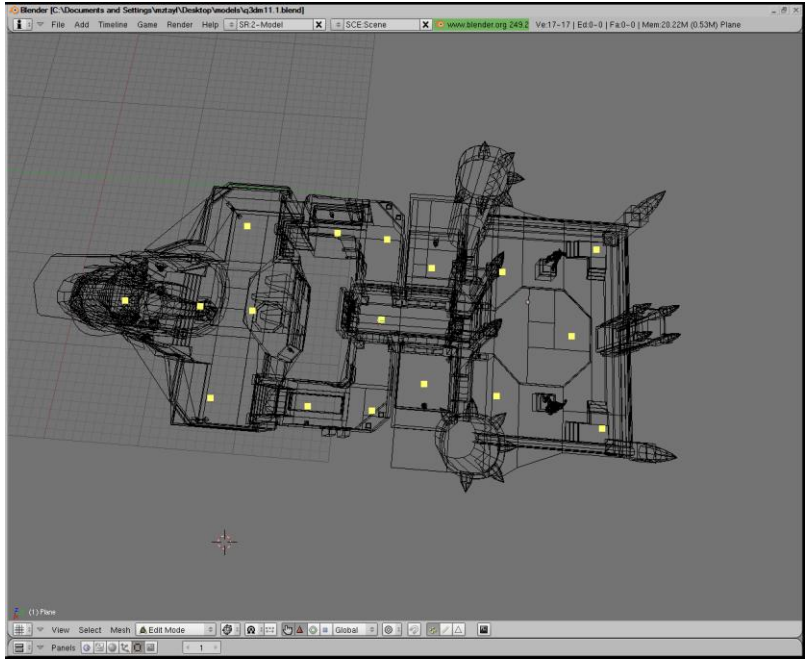
- Build a reverberation map
  - Get the acoustic response for each point
  - Any sound propagation method can be used
    - e.g, ray-tracing

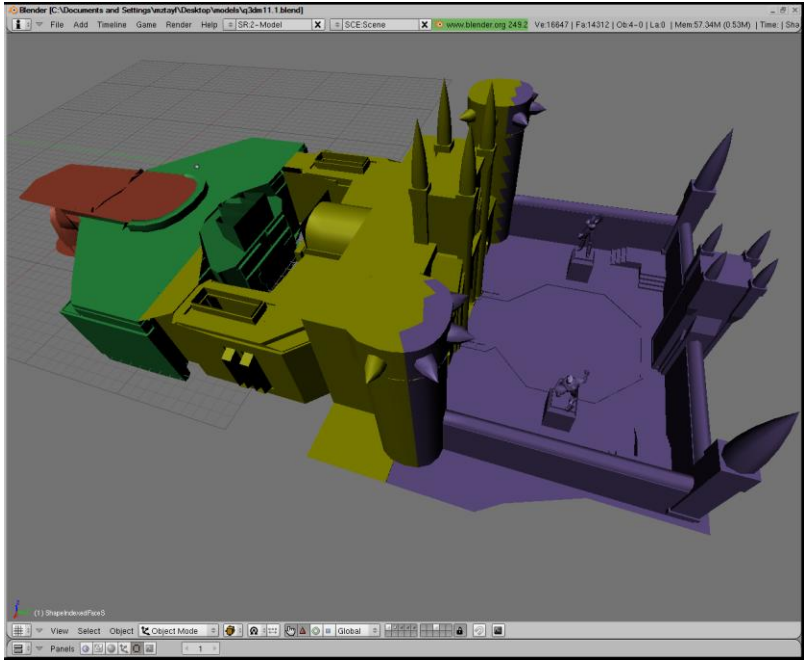




Precomputing geometry-based Reverberation effects for games
N. Tsingos, 35th AES conference, Audio for games, 2009

Modeling environmental effects





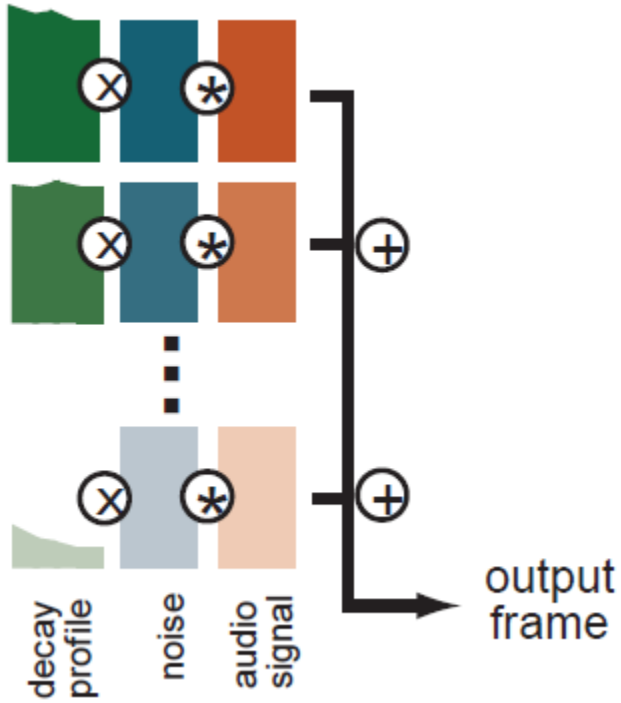
Rendering environmental effects

- For each listener and source
  - Find closest sample to listener*
  - Find closest sample to source*
  - Look up decay between the samples
  - Render source audio according to decay data

*in the same zone

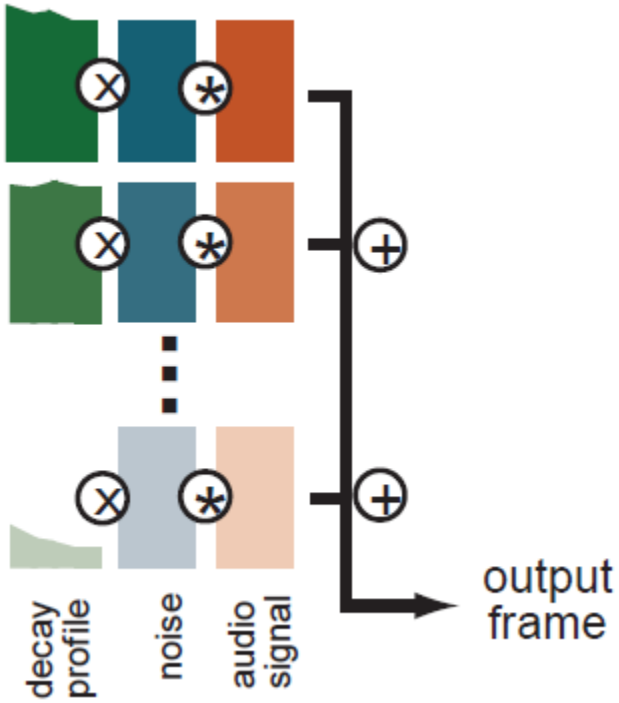
Implementation overview

- Lookup zone/sample point
- For each source
  - Get decay
  - Compute interpolation if needed
  - Convolve input audio with decay
  - Mix all audio streams

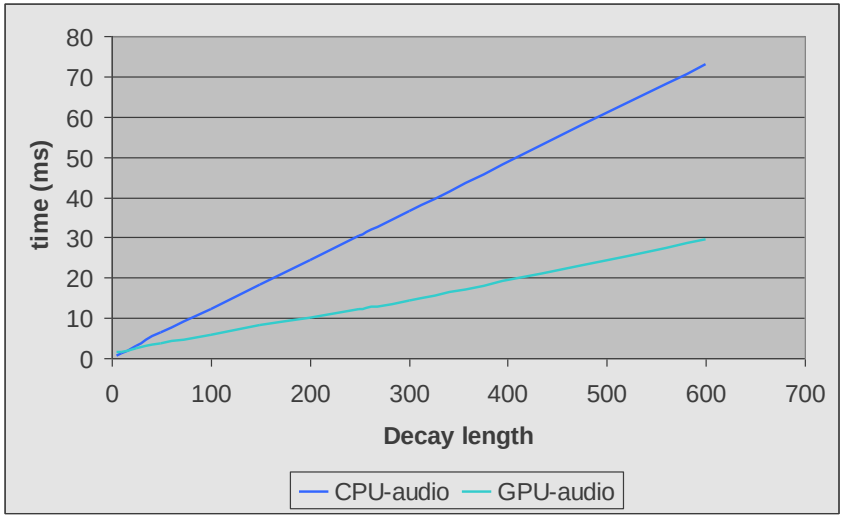


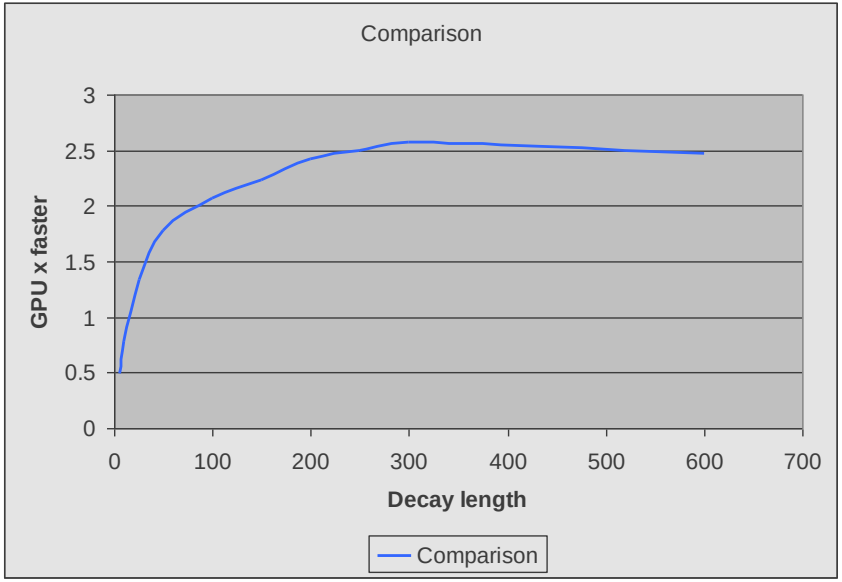
Implementation overview

- Lookup zone/sample point
- For each source
  - Get decay
  - Compute interpolation if needed
  - Convolve input audio with decay (GPU)
    - Must send data to GPU
  - Mix all audio streams



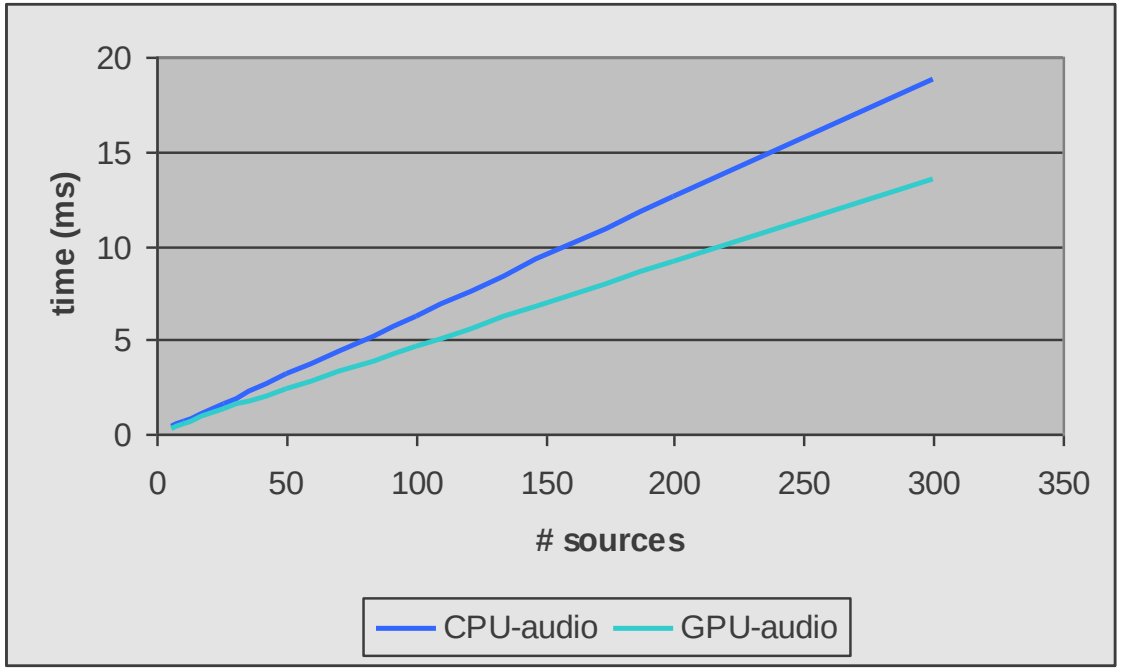
Performance – reverberation length





320 samples @ 16KHz, MDCT domain processing

Performance – # sources



Discussion and improvements

- GPU improvements
  - Reduce the required data transfers
  - Move more calculations to GPU
    - Decay interpolation
  - Increase frame size
    - Process 960 samples in each thread block on Fermi
  - Use constant memory
  - Build work queues to process large chunks of data

- **Large speed-up**
  - **x4.75 on Quadro FX5800 , x33 on Fermi (mix rate : 3+ million frames / sec !)**

More GPU for audio processing

- **Fast Fourier/Cosine transforms**

  audio coding, filtering [Govindaraju et al., GPUFFTW]

- **FIR filtering/convolution**

  reverberation, EQ [Siltanen et al., Cowan et al., Smirnov et al.]

- **Synthesis**

  modal synthesis of contact sounds [Trebien et al., Zhang et al.],
  sinusoidal additive synthesis [Savioja et al.]

- **Production tools and Digital Audio Workstations**

  GPU-accelerated VST plugins , Adobe Flash 10 supports GPU audio processing

GPU for numerical acoustics

Use of GPU for solving the wave equation, linear algebra

Finite element simulations

- finite differences, digital waveguide mesh
  [Röber et al., Micikevicius]
- frequency domain approach of [Raghuvanshi et al.]
  - x9 to x300 speed up reported

Radiosity

- use GPU for final gathering [Siltanen et al.]

GPU for geometry-based acoustics

Ray-tracing, occlusion, surface integrals

- Process geometrical primitives
- Can be used online or offline

Closer in spirit to graphics rendering problems

- fits well into the graphics pipeline
- 10x speed-up reported for GPU-based raytracing
  [Jedrzejewski et al., Röber et al.]
- can benefit from available APIs, e.g. Optix

GPU for scattering simulations

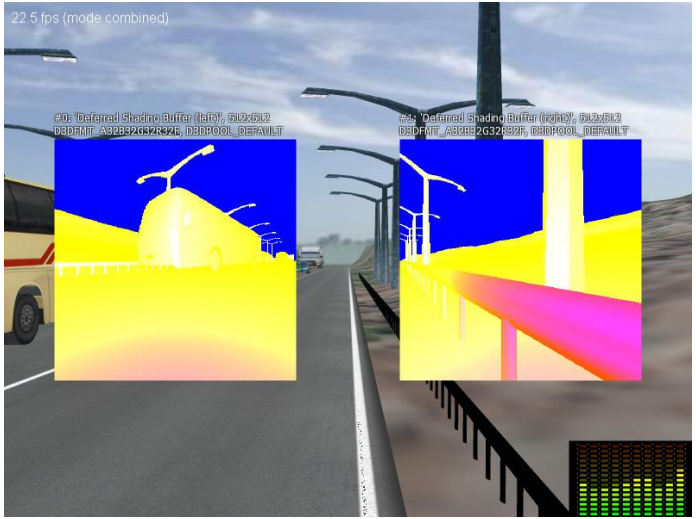
Physically-based modeling of scattering

- Unifies reflection, occlusion, diffraction
- Supports detailed, dynamic geometry

Kirchhoff approximation (KA)

KA is well suited to GPUs

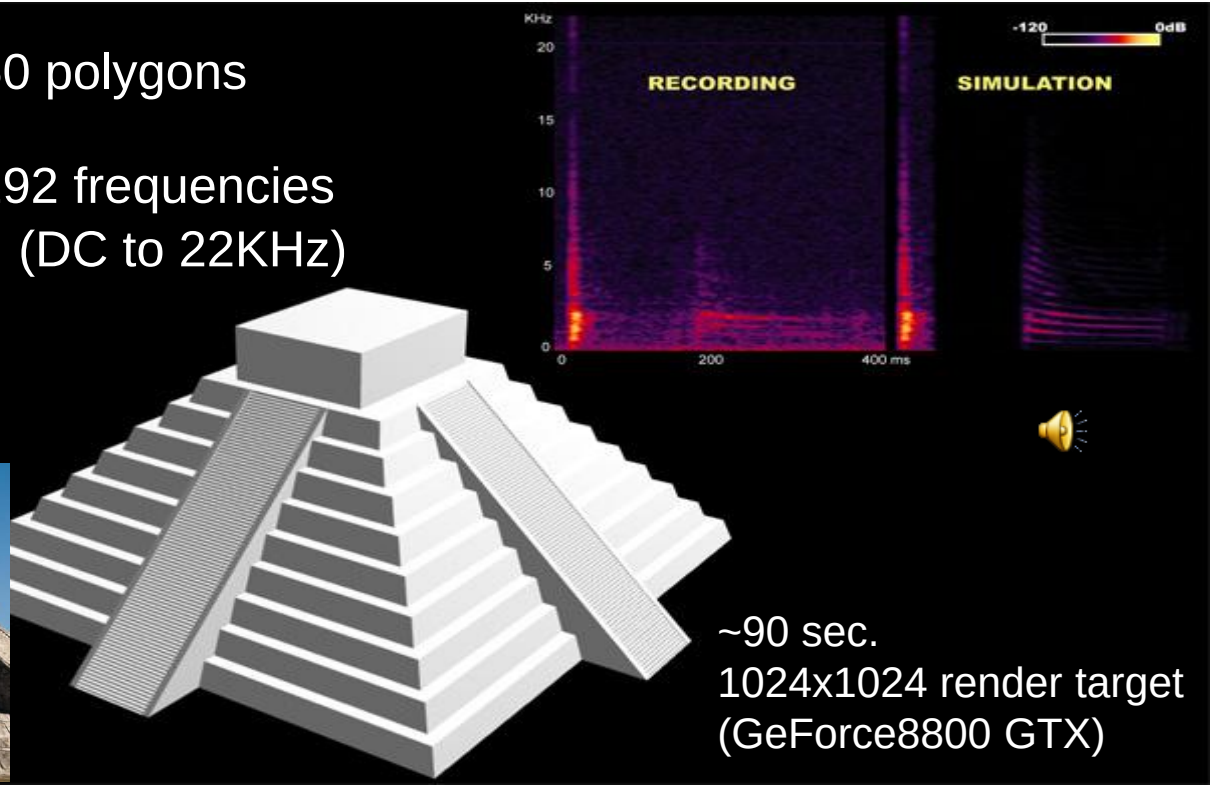
- Sample visible surfaces from sources
- Efficient integrand evaluation for each sample
- Hierarchical integration



Offline simulation: Kukulkan temple

860 polygons

8192 frequencies
(DC to 22KHz)

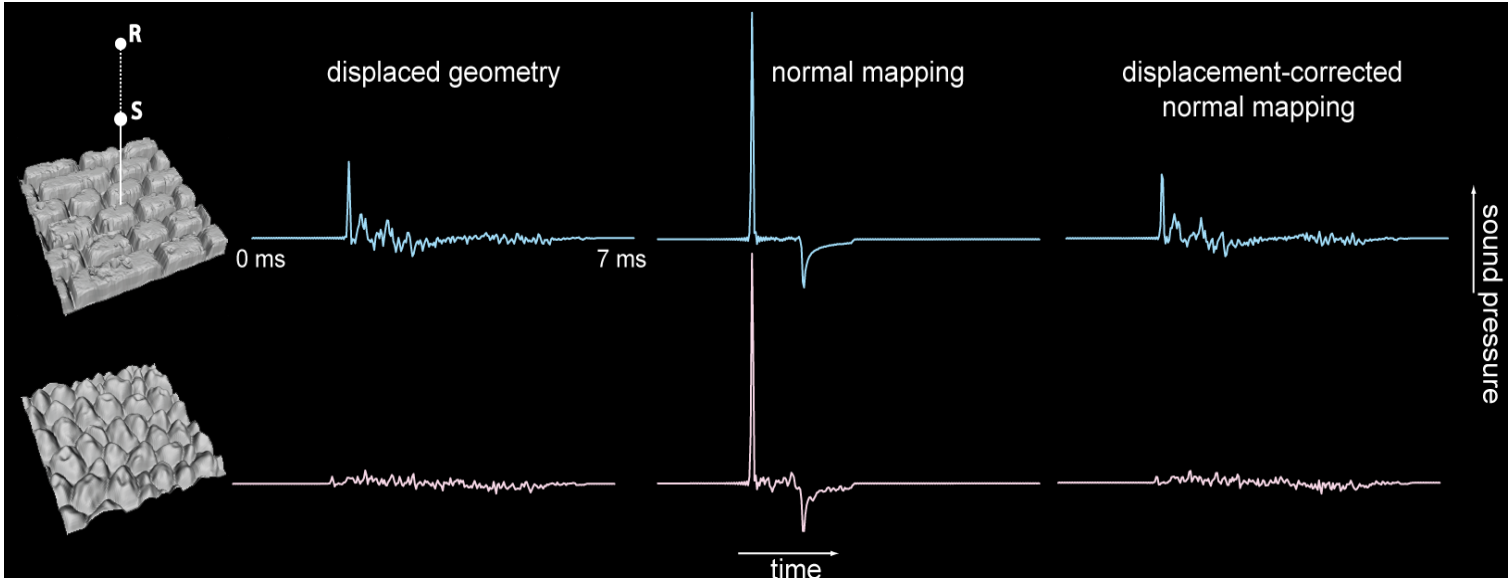


~90 sec.
1024x1024 render target
(GeForce8800 GTX)

Level of detail for sound scattering

Simplify geometry but preserve details in textures

[Cook et al. 87, Cohen et al. 98, Baboud et al. 06]

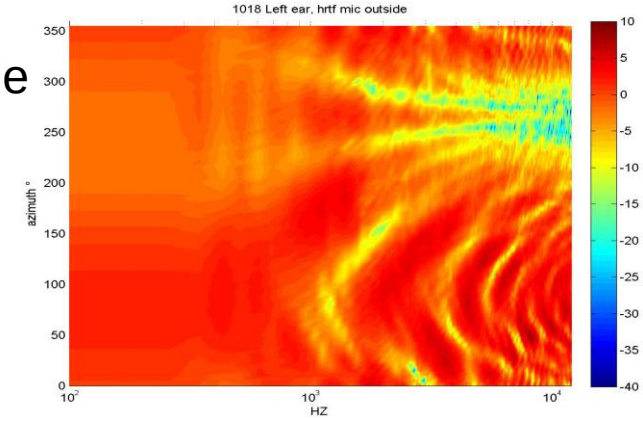


Individualized HRTF modeling

- **Head Related Transfer Functions**
  - Encode scattering from head,pinna and torso
  - Are used to reproduce 3D audio over headphones
  - Are unique to every listener

© IRCAM

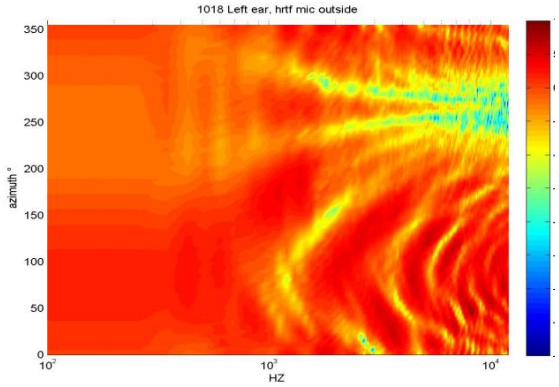
angle

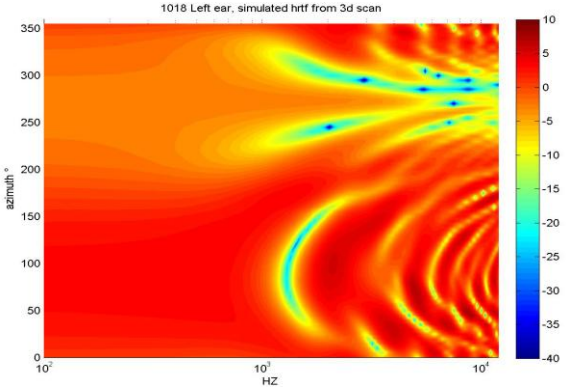


frequency

Individualized HRTF modeling

- **GPU for individualized model-fitting from photographs**
- **GPU for scattering simulation**
  - minutes instead of hours or days





measured **simulated**

courtesy of Olivier Warusfel and Van Nguyen, IRCAM

Limitations of GPUs for audio processing

- **Recursive filters**
  - Require non local access
- **Codecs**
  - Require sequential bitstream packing/unpacking
- **Ratio of compute to memory access is generally low(er)**
  - Frame sizes must be large enough but this implies latency
  - Optimal if dealing with many streams stored in graphics memory

Conclusions

- **GPU can be successfully used for acoustics**
  - audio processing
  - numerical acoustics
  - geometry processing
- **Speed-up : 2x to 300x**
  - Free-up CPU resources
- **Improved interactive simulations**
- **Support complex geometry with more accurate propagation models**

What's next ?

- **GPUs, programming tools and GPU-CPU interaction are getting better and better !**
- **Opens many applications for audio and acoustics**
  - acoustic reflectance functions
  - goal-based acoustic design
  - sound for physics-based simulation in games
  - microphone / loudspeaker arrays, Wave-field synthesis
  - speech recognition
  - seismics
  - audio for mobile devices

Thank you !

Manuel Asselot, Carsten Dachsbacher, Matteo Dellepiane, Emmanuel Gallo, Sylvain Lefebvre, Van Nguyen, Micah Taylor, Olivier Warusfel and Ian Williams from nVidia

See this paper for additional references :

Using Programmable Graphics Hardware for Acoustics and Audio Rendering
N. Tsingos. 127th AES convention (October 2009), Paper Number:7850

Visit : http://www-sop.inria.fr/reves/Nicolas.Tsingos

for related papers, results, movies

Questions? Email : nicolas.tsingos@dolby.com