Real-Time Room Acoustics Planning

Sönke Pelzer, Michael Vorländer
Institute of Technical Acoustics, 52074 Aachen, Germany, Email: spe@akustik.rwth-aachen.de

Introduction

Over the last decades computer-based room acoustical simulation has emerged to be a powerful tool in the architectural planning process. It enables to facilitate decisions on room shape, material selection and optionally on the design of a sound reinforcement system. However, commercial room acoustics simulation software is mainly used by a small group of experts, such as acoustic consultants and researchers. At the same time new acoustics algorithms have evolved in the domain of Virtual Reality (VR), where it is desired to reproduce virtual sound events as realistically as possible.

Those state-of-the-art real-time simulations already achieve accurate processing, a high reproduction quality and interactive update rates, so that they are highly interesting for application in classical domains (acoustic consulting, research) and could also pave the way for architects to directly integrate acoustics into their work flow. Applying modern interactive room acoustics simulation during the architectural planning process gives the architect complete control on the acoustical consequences of the room design during the early modeling stage and throughout the whole design process, even if only a trial-and-error paradigm is applied. To integrate acoustics into the workflow of architects, it is necessary to provide appropriate interfaces between room acoustics simulation and CAD modeling software.

This article describes the key components of room acoustics simulations to achieve interactive update rates, followed by interfaces such as plug-ins for typical 3D CAD room modeling tools, a validation of the used simulation algorithms and finally a selection of example applications opening new possibilities.

Challenges

Room acoustical simulations tools are developed since the late 1960s, when Krokstad et al. started using the ray tracing technique to predict acoustical room responses [2]. Although the available computation power has significantly improved since then, ray tracing is still computationally too expensive to be performed in real-time with high update rates. This holds true for both acoustical and visual rendering techniques. However, because for the time- and latency-critical early part of an impulse response, much faster methods are available (such as the image source method [3]), modern interactive systems include acoustical ray tracing for the late part of the impulse response[8, 4].

The challenges of such interactive room acoustical simulations are to harmonize physical accuracy, perceptual demands and algorithmic complexity. An elaborate analyze and control stage has to react appropriately on user interaction. The computational cost incurred if a virtual scene is subject to modification may vary significantly, dependent on the type of modification. This is further illustrated in Figure 2, which indicates the necessary computations for a certain scene change. In Figure 1 a basic overview is provided how the computational cost rises when elements of different type are subject to modification. One of the cheapest operations is to update the sound reproduction in case of a rotation of the receiver, a typical case when using head-tracking. On the other side, the most expensive operation is a change to the geometric structure of the virtual scene. It requires a complete recalculation of the sound propagation and impulse responses, and prior to that a significant time to rebuild acceleration structures for the simulation algorithms, which are outlined in the next section.

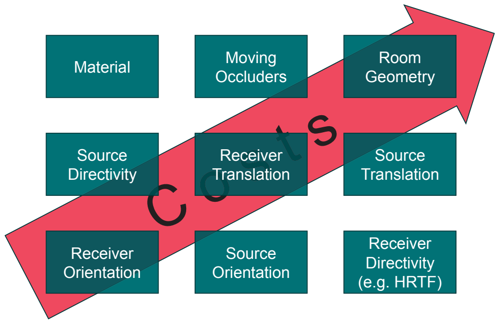

Figure 1: Challenges in interactive room acoustics simulation. Head-tracking (receiver orientation) can be implemented very efficiently, while changes to the geometrical room structure are very challenging in real-time.

Object	Event	Image Sources			Ray-Tracing	Filter-synthesis
		Build	Translate	Audibility		
Receiver	Rotation					•
	Translation			•	(•)	•
	Directivity					•
Source	Creation	•		•	•	•
	Rotation					•
	Translation		•	•	(•)	•
	Directivity					•
Room	Material				•	•
	Geometry	•		•	•	•

Figure 2: Possible interactions with the virtual scene require different recalculations.

Acceleration

The most frequently used operation in geometrical room simulations are intersection tests between rays (representing the sound propagation) and polygons of the scene geometry. This operation is responsible for nearly the all the calculation time, so that an acceleration can significantly improve the overall performance. The introduction of spatial data structures can decrease the computation time of each single intersection. The speed-up is dependent on the complexity of the spatial data structure. Typically, a more complex data structure also offers high performance gain. The following algorithm classes are sorted descending by complexity and performance:

- Binary Space Partitioning
- kd-Trees
- Octrees
- Bounding Volume Hierarchies
- Spatial Hashing

However, the low complexity of the latter algorithms, such as spatial hashing, has the advantage of much faster rebuild times. Former approaches aimed at separating static and dynamic geometry or separating between modification and listening phases to apply different spatial structures (binary space partitioning (BSP) and spatial hasing (SH)) at the same time [12]. Both ideas implicated a restriction in terms of full user interaction. Therefore, a recent suggestion was to use SH only for the early part of the impulse response (low reflection order) and BSP only for the late part [4]. Using this hybrid method offers high update rates for the perceptually prominent early part, while using the highest performance for the ray tracing part, which requires orders of magnitude more intersection tests. An exemplary simulation activity is depicted in Figure 3. The Bounding Volume Hierarchies (BVH) can be seen as a compromise solution, offering fast rebuild times, while still having competitive performance.

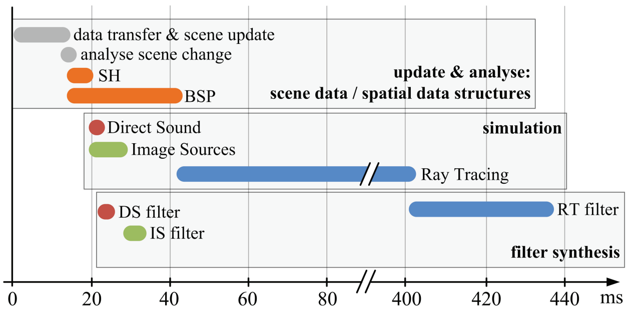

Figure 3: Simulation sequence after an interaction event, with a detected room geometry change. Direct sound and early reflections are updated with very low latency using the flexible spatial hashing technique, while the costly ray tracing waits for the high-performance binary space partitioning structure.

In ray tracing algorithms for computer graphics, the use of SIMD instructions provided further performance gains. Because these rely on consistent ray coherence, they are not likely to perform well for acoustical ray tracing, where scattering breaks up ray packets and typical reflection orders are much higher. Lately, hybrid strategies that used BSP and BVH have been proposed to overcome this problem [6]. However, to exploit the resources of modern CPUs, it is necessary to execute code in parallel. The ray tracing technique is well suited for parallel execution, because of minimal data dependencies between threads, so that nearly perfect scalability to CPU cores or cluster nodes have been reported [8, 9, 10].

Real-time room acoustics implementation

This section introduces a recent implementation of a state-of-the-art real-time room acoustics simulation. Developed at the Institute of Technical Acoustics at RWTH Aachen University, it takes advantage of hybrid geometrical acoustics algorithms, hybrid spatial data structures, and multi-layer parallelism (node-level, multi-core, SIMD) [8, 7, 9]. This enables the application for real-time simulation, visualization, and auralization including low-latency convolution for 3-D reproduction via loudspeakers or headphones. Supposing that enough computation power is available, large virtual scenes can be simulated, including effects such as directivities, doppler shifts, scattering, higher-order edge diffraction, and sound transmission through walls. The room geometry, material data, and acoustical source/receiver characteristics can be modified at run-time through manifold interfaces that are introduced in the next section. The reproduction module renders source signals by convolution with simulated spatial 3-D impulse responses, with signal processing for binaural headphones, crosstalk-cancellation, multi-channel intensity panning and higher-order Ambisonics. The spatial impulse responses can also be composed in a hybrid manner using multiple reproduction techniques [11].

Interfaces

A room acoustical simulator is traditionally operated by using a graphical user interface (GUI). However, for real-time application it is favorable to access the simulation core directly using a network protocol, such as TCP/IP, or linking the library directly. This requires a dedicated master application, for example an existing virtual reality framework [12].

```
rpf = RavenProject('Jazzclub.rpf');

rpf.setReceiverPosition([4.1, 1.7, 2.3]);

rpf.setMaterial('Stage_Back_Wall', 0.6, 0.2);

rpf.run();

hoa = rpf.getAmbisonicsImpulseResponse();

Drums3D = rpf.convolve('Drums.wav', hoa(1,1));
Bass3D  = rpf.convolve('Bass.wav', hoa(2,1));
```

Figure 4: Example code of a MATLAB script, showing a receiver position translation and a material property change with simple commands. Finally the an auralization is prepared using convolution.

In research it is often desired to run scripted sequences, e.g. preparing many variations, analyzing influence of a certain parameter, preparing series of auralizations for listening tests, or preparing animations. Therefore a direct link to a scientific programming language, such as Python or MATLAB, must be established. In Figure 4 an example code of a controller interface for MATLAB is shown, illustrating how the receiver position or a material's absorption can be changed easily with a final auralization using convolution.

Considering that a room simulation cannot be done without prior preparation of the 3D CAD room model, the geometrical input is one of the crucial parts of the user interface. If the geometry is about to be modified interactively, for example during an architect's design process, this CAD modeling functionality must be well engineered and user-friendly. The fact that very comfortable 3D modeling programs already exist and are frequently used suggests that integrating the acoustics simulator into such modern programs makes perfect sense. As a prove of concept, our real-time room acoustics renderer was integrated into *Trimble SketchUp* [13, 4], as shown in Figure 6.

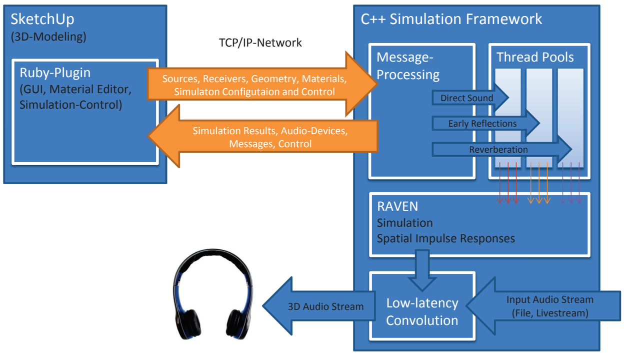

Figure 5: Using a plug-in for the CAD modeler *SketchUp*, all relevant scene information is transmitted over network to the room acoustics simulation in real-time. The multi-threaded simulation renders 3-D audio and returns results for visualization (e.g. room acoustics parameters) back to *SketchUp*.

The connection between *SketchUp* and the acoustics simulator was achieved by enabling a network connection between the two programs, as shown in Figure 5. A plug-in for *SketchUp* was developed in the programming language *Ruby*, implementing a TCP/IP server that transfers geometry data including materials, as well as sound sources, receivers, and the simulation configuration to the acoustics simulator in real-time. The simulator deals with further processing and sound reproduction and provides a return channel, which is used to transfer visual results back to *SketchUp*. Standard room acoustics parameters, such as reverberation time, clarity, strength, and sound propagation paths (e.g. early reflections) can be visualized directly inside *SketchUp* and are updated on-the-fly.

With an auralization module that calculates spatial impulse responses and renders 3-D sound with realistic room acoustics in real-time during the modeling process, *SketchUp* is extended to provide full room acoustics feedback. The reproduction module supports binaural (headphones or crosstalk cancellation) and array rendering (VBAP and Higher-Order Ambisonics). Feeding the low latency convolution from the live input of the sound card, the user can talk into the virtual scene with a microphone, while actively modeling other sound sources, materials and geometry at run-time. For more complex compositions of multiple sound sources, e.g. in soundscape applications, the audio signals can also be transferred from typical digital audio workstations using the virtual studio technology (VST) [14].

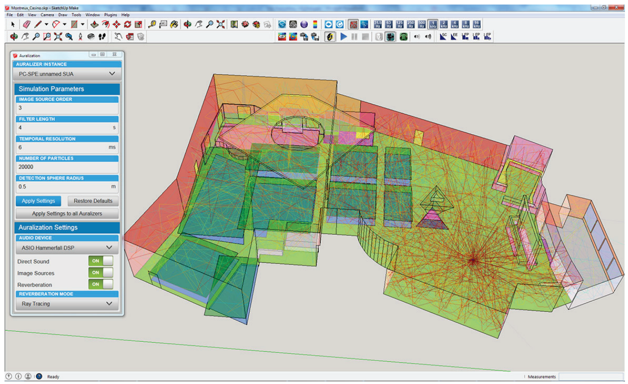

Figure 6: Computation times for early reflections based on two spatial hashing traversal schemes (VC and VT), for different room models. The voxel size is varied in relation to the average edge length of the scene's polygons. A minimum is found for approximately 1.5 times the average edge length.

Performance

A detailed performance analysis of a state-of-the-art real-time implementation would be beyond the scope of this paper, but can be found in literature [7, 8, 10]. On typical commodity hardware (Quad-core CPU, 2.6 GHz), it is possible to simulate/auralize about 10 individual sound sources, while achieving interactive update rates for the important direct sound (>30Hz) and early reflection filters (>3Hz), and while simultaneously performing ray tracing (reverberation filter updated every 4 seconds for each of the 10 sound sources) [10]. Performance measurements for 1-18 sound sources are shown in Figure 7.

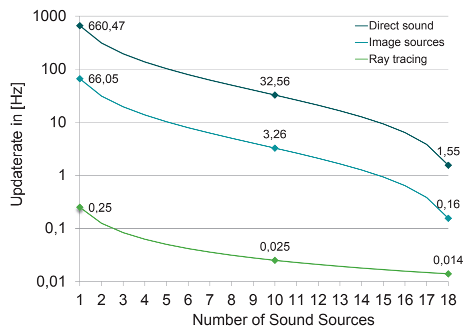

Figure 7: Simulation performance on a Quad-core CPU with simultaneous real-time rendering of multiple sound sources. Interactive update rates can be maintained up to ca. 10 individual sound sources including streaming convolution.

Validation

By comparing the simulation results of the presented library with measurements of a real rooms, the implemented algorithms were validated. One test case were a reverberation chamber with more than ten seconds of reverberation at low frequencies. Using a two-way studio monitor and an omni-directional microphone, impulse responses were measured and simulated at several positions. The spectrogram of one measurement position is shown in Figure 8. Although the quantization in 31 third-octave bands is visible in the simulated response, the overall match is very good. Further validation results can be found in literature [15, 8].

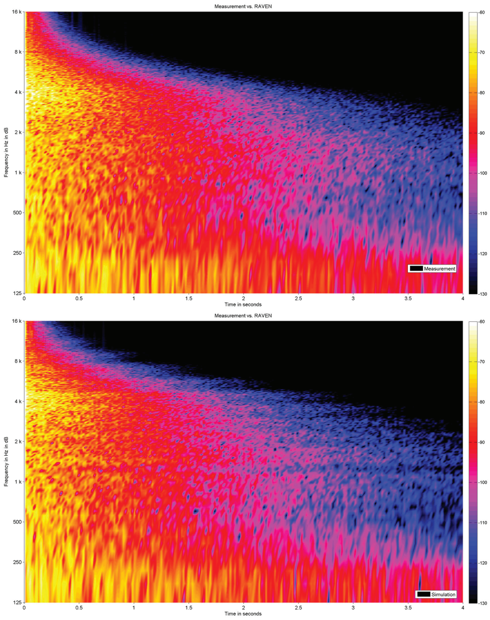

Figure 8: Comparison of the spectrograms of measured (top) and simulated (bottom) impulse responses of a reverberation chamber.

Conclusion

This contribution gave insights about recent developments in room acoustics simulation. A big challenge is interactive geometry, which is needed in virtual reality applications and interactive room acoustics design. A solution was presented by the introduction of a hybrid spatial data structure, providing a flexible spatial hashing algorithm for low latency update of direct sound and early reflections, and high-performance binary space partitioning for efficient calculation of the multitude of reflections in the late decay. Using such advanced techniques in combination with extensive parallel computing, i.e. multi-threading and vectorization, realizes a fully interactive real-time room acoustics simulation.

To increase the utilizability of such modern simulation algorithms, they must provide interfaces that enable common users to control and access them. A researcher most likely uses a scientific programming language, e.g. MATLAB or Python, while an audio engineer is accustomed to digital audio workstations, so that interfaces to such software were presented. With the focus on room acoustics planning, the users are architects, civil engineers, and acoustic consultants. Therefore an interface to the popular CAD modeling tool *SketchUp* was introduced, offering a seamless integration of room acoustics rendering into its main program surface. With permanent live acoustics feedback during architectural work, it can be expected that typically predominantly visually driven design decisions experience an increased attention towards acoustical aspects.

The requirements therefore are seamless integration, user-friendliness, high performance, and easy understanding of the results. To the experience of the authors, the understanding of room acoustical phenomena is highly enhanced by the auralization technique, making effects directly audible in addition to the visual monitoring of room acoustics parameters.

References

[1] Cadet, G. & Lecussan, B.: Coupled use of BSP and BVH trees in order to exploit ray bundle performance. IEEE/EG Symposium on Interactive Ray Tracing, 2007

[2] Krokstad, A., Strom, S. & Sørsdal, S.: Calculating the acoustical room response by the use of a ray tracing technique. Journal of Sound and Vibration 8(1), 118-125, 1968

[3] Allen, J. B. & Berkley, D. A.: Image method for efficiently simulating small?room acoustics. The Journal of the Acoustical Society of America, 65(4), 943-950, 1979

[4] Pelzer, S. et al.: Interactive real-time simulation and auralization for modifiable rooms. Building Acoustics, 21(1), 65-74, 2014

[5] Schröder, D., Ryba, A., & Vorländer, M.: Spatial data structures for dynamic acoustic virtual reality. In Proc. 20th Intern. Congress on Acoustics, Sydney, 2010

[6] Wald, I. et al.: Interactive rendering with coherent ray tracing. Proceedings of Eurographics 20, 153-164, 2001

[7] Pelzer, S. et al.: Integrating real-time room acoustics simulation into a CAD modeling software to enhance the architectural design process. Buildings, 4(2), 113-138, 2014

[8] Schröder, D.: Physically based real-time auralization of interactive virtual environments (Vol. 11). Logos Verlag Berlin GmbH, 2011

[9] Wefers, F. et al.: Interactive acoustic virtual environments using distributed room acoustic simulations. Proc. EAA Joint Symp. on Auralization and Ambisonics, 2014

[10] Schallenberg, R.: Multi-core Parallelism for Real-time Room Acoustics Simulation. Master's thesis, RWTH Aachen University, 2015

[11] Pelzer, S. et al.: 3D reproduction of room auralizations by combining intensity panning, crosstalk cancellation and ambisonics. Proc. EAA Joint Symposium on Auralization and Ambisonics, 2014

[12] Schröder, D. et al.: Virtual reality system at RWTH Aachen University. Proc. International Symposium on Room Acoustics, Melbourne, Australia, 2010

[13] Trimble SketchUp Homepage, URL: `http://www.sketchup.com/`

[14] Steinberg Virtual Studio Technology, Plug-In Specification 2.0, SDK. Documentation Release #1, 1999

[15] Pelzer, S. et al.: Quality assessment of room acoustic simulation tools by comparing binaural measurements and simulations in an optimized test scenario. Proc. Forum Acusticum, Aalborg, Denmark, 2011