Pre-computing geometry-based reverberation effects for games

Nicolas Tsingos[1]

[1]*Dolby Laboratories, San Francisco, CA, USA*

Correspondence should be addressed to Nicolas Tsingos (`nicolas.tsingos@dolby.com`)

ABSTRACT

Current video games either pre-render reverberation effects into the sound effects or implement them at run-time using artificial reverberation filters. While interactive geometrical approaches can be used for more accurate acoustical modeling, the increased authoring complexity and the additional cost of geometrical calculations still appears to overshadow their potential benefits.
This paper presents solutions to integrate off-line geometrical acoustic modeling in game environments. By pre-computing image-source gradients for early reflections and directional decay profiles, we can generate location dependent reverberation effects without storing or accessing the actual geometry at run-time. We render such reverberation effects using a frequency-domain scalable processing approach. In this context, we introduce an efficient prioritization scheme and evaluate alternative transforms for late reverberation processing. Our pipeline enables fine-grain rendering of distance and surface proximity effects and modeling of both outdoor and coupled indoor spaces with arbitrary reverberation decay profiles.

1. INTRODUCTION

Sound reverberation effects due to sound scattering off wall surfaces carry major cues related to the size of the environment and distance to sound sources [1]. Therefore, reverberation helps users to establish a better sense of presence in virtual environments and is arguably one of the most important audio effects to simulate in gaming applications.

In current video games, reverberation effects are either directly pre-rendered into the sound effects or implemented at run-time using dynamic artificial reverberation filters [9, 6]. Parameters of the reverberation decay can be directly manipulated by the sound designer to achieve a desired effect without requiring any geometrical modeling. While simplifying the authoring process, traditional artificial reverberators suffer from a number of issues. They impose a "single room" model and constrain the shape of the decay profile (e.g., exponential). They make limited use of geometry and therefore fail to convincingly model coupled or outdoor spaces or provide finer-grain surface proximity effects. Finally, they do not scale to accommodate large numbers of concurrent effects. While a number of geometrical approaches have been presented to model dynamic sound reflection and diffraction interactively [12, 14, 5], geometrical acoustics never really found their way into gaming applications; the increased authoring complexity and the additional cost of geometrical calculations still appearing to overshadow their potential benefits.

This paper presents solutions for integrating geometry-based acoustic modeling in a game environment without storing or accessing the actual geometry at run-time, thus requiring little computational resources and a low memory footprint. We show how geometrical reverberation authoring can easily fit within the graphics authoring pipeline and how geometrical calculations can be performed off-line, thus enabling efficient run-time processing. Specifically, given a 3D model of the environment, we propose to pre-compute sound reflections off surfaces at a number of key-locations using traditional ray-tracing approaches [27]. We extend previous approaches to pre-compute image source gradients and hybrid directional-diffuse energy decay profiles and introduce compact data structures to share common information across key locations. A benefit of this approach is that the run-time complexity does not directly depend on the geometrical complexity of the input 3D model. At run time, the gradients are used to generate the appropriate set of image sources for arbitrary source locations without requiring explicit access to the geometry.

To support large numbers of sources, we further improve upon previous frequency-domain scalable processing approaches [25, 23] by introducing a more efficient masking algorithm and a faster Modified Discrete Cosine Transform (MDCT)-domain progressive processing of late reverberation effects. The resulting pipeline can process more than a hundred concurrent sources on a single PC core with high quality, dynamic environmental effects. It enables fine-grain modeling of distance and surface proximity effects and modeling of both outdoor and coupled indoor spaces with arbitrary reverberation decay profiles.

2. RELATED WORK

A large body of work has been devoted to simulating sound propagation effects in virtual environments. In the following section we review the techniques most directly related to our work.

2.1. Statistical acoustics

Statistical acoustics (SA) are widely used to determine energy decay rates in rooms and lead to efficient approaches to model reverberation effects. For instance, the Eyring model describes the energy decay in an enclosure as a decaying exponential:

$$E(t) = E_o e^{\frac{cS}{4V} t \log(1-\bar{\alpha})}, \tag{1}$$

where c is the speed of sound ($\approx 340 m.s^{-1}$), V is the volume of the space, S the total surface area of the walls and $\bar{\alpha}$ the surface-averaged absorption coefficient.

While such approaches can give good results in the case of single rooms, they fail to capture the multi-slope decays created when rooms are connected. SA methods specifically designed to predict reverberation time in coupled volumes also exist but assume that reverberant decays for each room in the coupled system are exponentially decaying and can be predicted independently.

A solution to more accurate modeling is to use methods based on geometrical acoustics.

2.2. Geometrical acoustics

Geometrical acoustics (GA) is probably the most popular approach for physically-based acoustic modeling. GA is a high-frequency approximation that models sound propagation along ray-paths. Early reflections are usually modeled using virtual *image sources* (IS), i.e., discrete delayed and filtered copies of the original source signal (see Figure 2). An IS corresponds to a mirror (specular) reflection path off the surfaces of the environment. For a given source position, a large number of IS have to be recursively constructed to obtain a convincing reverberation filter. Specular reflection paths can be constructed using techniques such as ray or beam tracing [12, 14, 5]. Monte-Carlo path tracing or radiosity-like approaches can be used for diffuse exchanges [8, 3, 21].

When sources are moving, image sources need to be updated. Depending on the location of the source and listener, different sets of reflections can be valid, also requiring visibility checks. In a recent paper [14], Lentz et al. report update times of up to 3 sec. to update full reverberation filters in a Monte-Carlo path tracing context. Early reflection visibility checking takes about 0.7 sec. for up to order 3 in their 105 polygon test model, leading to 300000 potentially visible image sources to test. While acceptable in the context of acoustical design, such update rates are limiting for typical consumer entertainment applications, such as games, where the simulation should typically run between 10 and 60Hz to accommodate fast listener or source motions and geometrical changes in the environment. Another paper [11], using a beam-tracing approach, reports constructing all third order reflections of a sound source in 1 msec. for a 1190 polygon model. However, even the fastest interactive GA approaches remain limited to low-orders of reflection (typically 8) and very few sources. They cannot be applied to compute full reverberation effects interactively since reverberation filters typically contain audible reflections of order 100 or more.

To avoid the cost of an on-line GA simulation, Pope et al. [17] have proposed to pre-compute and store the reverberation filters for a number of locations in a virtual environment, at the expense of a large memory footprint. In this paper, we propose to pre-compute costly GA-based solutions in an off-line process and present compact alternatives to storing the full reverberation filters. We also propose solutions in order to support interactive modifications of the environment.

2.3. Artificial reverberators

In typical indoor environments, an exponential number of echoes reaches the listener as the reflection order increases. The later part of the reverberation is then perceived as a continuum and tends to be more spatially diffuse and slowly varying throughout the environment. Statistical acoustics formulations lead to the

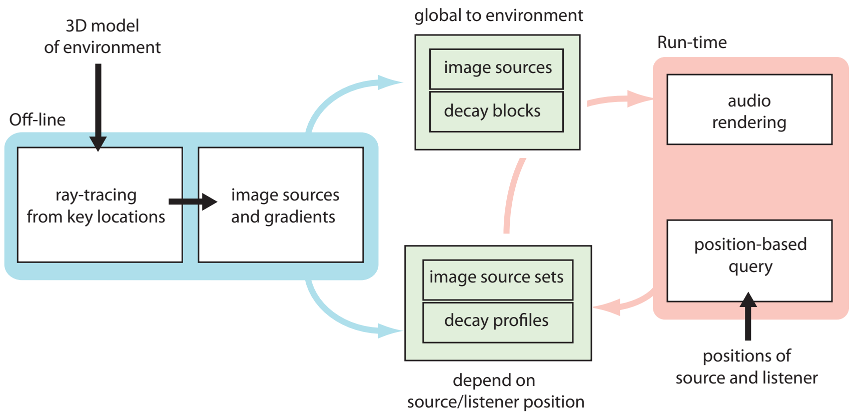

Fig. 1: Overview of our approach. In a pre-processing step (left), we compute geometry-based image sources and their gradients and derive reverberation decay profiles for a number of key locations in the environment. At run-time (right), a suitable combination of image sources and decay profile is retrieved for each source-listener pair. Image sources are interpolated from the key position to the actual source position using the pre-computed gradients.

development of efficient *artificial reverberators*, which widely used to auralize late reverberation effects in games [6, 18]. Artificial reverberators do not model the fine-grain temporal structure of a reverberation filter but assume that reverberated components can be modeled as a temporal noise process modulated by slowly-varying energy envelopes in different frequency sub-bands. Following Eq 1, these envelopes are often considered as exponentially decaying, which lead to the design of efficient recursive *Feedback Delay Network* (FDN) filters [20, 9, 6, 18]. Alternative frequency-domain approaches have also been proposed [26]. This approach of modeling the fine-grain structure of reverberation filters as a noise process has also been used to auralize Monte-Carlo path tracing results [10, 19]. In this case, a short-time integrated impulse response is constructed by path-tracing and is used to modulate the noise process. In the general case, this approach cannot be directly implemented using the previous FDN formulation due to non-exponential decays, and results in a high computational cost. In this work, we will also be using this latter solution but show that it can be efficiently implemented within a scalable Fourier-domain framework [25]. We also extend energy decay profiles to include directional information, which can be used to model directional echoes commonly found in outdoor scenes.

3. OVERVIEW

Figure 1 shows an overview of our approach. Similar in spirit to pre-computed light maps or radiance transfer [22] for computer graphics rendering, we generate reverberation effects off-line and interpolate them at runtime according to the position of the sources and listener. The architecture of the proposed reverberation engine follows the classical approach of separating early reflections from late reverberation. In section 4, we show how ray-tracing can be used to pre-compute image sources and reverberation decay profiles at key locations across the environment. We introduce compact data structures for efficient run-time access. In particular, we decouple the image sources from their visibility using *indexed image-source sets*. In addition to the position of the image sources, we also pre-compute their gradients so they can be extrapolated at run-time for arbitrary source locations. Similar to artificial reverberators, we only control the late reverberation using the time-frequency envelope of the energy decay and model its fine-grain temporal structure as noise directly at render-time. The time-frequency envelope is also pre-computed from the obtained image-sources. In section 5, we show how to use the pre-computed information to generate high-quality dynamic reverberation effects when sources or the listener are moving throughout the environment. Section 6 presents a scalable processing pipeline to render our dynamic reverberation effects and introduce several improvements to previous frequency-domain approaches. Finally, we discuss performance and benefits of our approach in Section 7 before concluding.

4. PRE-COMPUTING REVERBERATION DATA

Given a 3D model of the environment, sets of image sources and energy decay profiles are first computed off-line using classical GA approaches, such as ray-tracing [27] (see Section 2.2). A random set of rays is uniformly emitted in all directions from each candidate source location and recursively propagated until it reaches a receiving sphere at the candidate listener location. Our current implementation only considers specular reflections but could be easily extended to include diffuse (lambertian) scattering. As an alternative to a custom ray-tracer, acoustical modeling tools such as ODEON or CATT [16, 2] could also be used. In order to sample the environment, the ray-tracing must be performed for different key source and listener locations, whose positions can be specified by the user in a visual authoring tool. Once the N key locations are determined, the system generates N^2 lists of specular reflection paths considering all possible pairs of source and listener defined by the key locations. For instance, in the two-room environment of Figure 2 the user could specify 3 key locations, one at the center of each room and one at the portal and the system would compute 9 solutions assuming each location is, in turn, either a source or a listening point.

4.1. Image sources and derivatives

Each primary source S_k, creates multiple virtual image sources. During the ray-tracing step, we construct a global list of image sources and store their 3D position. We also pre-compute positional gradients encoding how the image source drifts when the key primary source S_k moves. For planar surfaces, this is equivalent to pre-computing the combined matrix of mirror-symmetry operators generated by successive specular reflections. For each virtual image source, $I(X,Y,Z)$, we construct the tensor H of partial derivatives according to the position of the primary source $S_k(x,y,z)$:

$$H = \begin{pmatrix} \frac{\partial X}{\partial x} & \frac{\partial Y}{\partial x} & \frac{\partial Z}{\partial x} \\ \frac{\partial X}{\partial y} & \frac{\partial Y}{\partial y} & \frac{\partial Z}{\partial y} \\ \frac{\partial X}{\partial z} & \frac{\partial Y}{\partial z} & \frac{\partial Z}{\partial z} \end{pmatrix} \quad (2)$$

This 3×3 matrix can be easily computed during the ray-tracing step by successively mirroring the source point S_k and three offset points $S_k^{x,y,z}$, displaced by 1 unit in each direction x, y and z (e.g, $S_k^x = S_k + (1,0,0)$). If $h_k^{x,y,z}$ are the mirror images of these three offset points, the lines of H are directly given by the three vectors $I - h_k^{x,y,z}$. Figure 2 shows a set of image sources computed at a highlighted key position and their positional gradients.

Using the matrix H, we can store the image source position I_0 assuming the source was at $(0,0,0)$, simply by offsetting I by the vector $-HS_k$. Storing all image sources relative to $(0,0,0)$ allows for sharing image-source structures across different locations in the environment for which the same image-sources are valid.

A frequency-dependent attenuation factor in decibel scale can also be stored with each image source to model the product of all surface absorption coefficients for different frequency sub-bands. An example data structure for the image sources, with quantized dB scale attenuation, is shown below.

```
struct ImageSource {
        float I0[3];
        float H[3][3];
        unsigned char attenuation[NBBANDS];
};
```

The number of image sources to store for an environment can be left as a user-defined parameter. Typically, the image sources will be sorted by increasing propagation time and the earliest ones will be kept for each source/listener pair.

4.2. Energy decay profiles

All image sources which are not explicitly stored will be modeled as part of the late reverberation by integrating their energy into *decay profiles*. Decay profiles are used to model an artificial reverberation filter as a temporal noise process windowed by a time-frequency decay envelope (see Section 2.3). Since decay profiles are usually smooth, they can be sampled at a low rate, typically 80Hz, and only require a limited memory footprint.

To build the decay profile corresponding to a source and listener configuration, we simply integrate the energy carried by each image-source according to its arrival time, which we quantize into a number of *decay blocks* at the desired sampling rate. If diffuse energy exchanges are modeled, the energy of each diffuse ray can also be directly integrated into the profile during the ray-tracing step.

Our decay block structure stores attenuation values (e.g., in dBs) for different frequencies to model the time-frequency envelope of the reverberation filter. We also

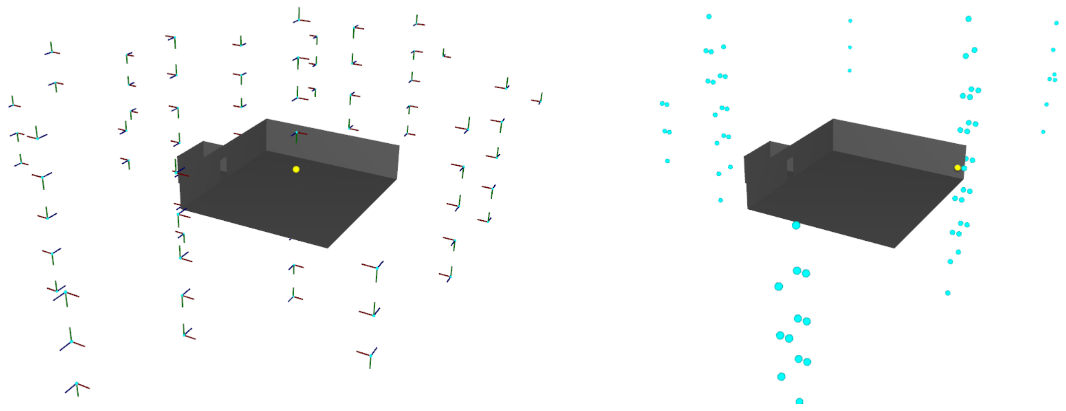

Fig. 2: Left: Example virtual image sources computed for a primary source located at the position of the yellow sphere. Directional gradients are also shown. Right: the same set of image sources (cyan spheres) extrapolated for another source position (yellow sphere).

compute and store a principal direction and a diffusiveness index that represents the ratio of directional-to-diffuse energy (e.g., 1 is pure directional, 0 is pure diffuse) within each decay block. The representative direction is computed as the energy-weighted average of all IS incident direction at the listening point. The diffusivity index is the length of that average vector. This is similar in spirit to the Spatial Impulse Response Rendering (SIRR) proposed in [15], providing a compact way of storing directional information. The principal direction is encoded in world-space and can be rotated to follow the orientation of the listener at run-time. This is useful for rendering echoes from portals (e.g., doorways) or from buildings in outdoor scenes which have stronger directional components. In first-person games, this is particularly important since the player can usually quickly look around. Figure 3 shows an example pre-computed decay profile and its diffusiveness index. An example data structure for the decay block (with quantized dB scale attenuation/diffusiveness and quantized direction on the sphere) is shown below.

```
struct DecayBlock {
        unsigned char direction;
        unsigned char diffusiveness;
        unsigned char attenuation[NBBANDS];
};
```

An energy decay profile is then simply encoded as a list of pointers (or indices) to *DecayBlock* structures. The primary motivation for storing decay profiles as a list of blocks is that a significant number of blocks can be vector quantized and shared amongst different decay profiles. Typically, this will result in different early blocks for various positions in the environment while sharing identical late blocks.

4.3. Image-source sets and validity checks

A significant number of image sources are going to be shared between different locations in the environment. We propose to store *indexed image-source sets* which are lists of visible image sources given a primary source

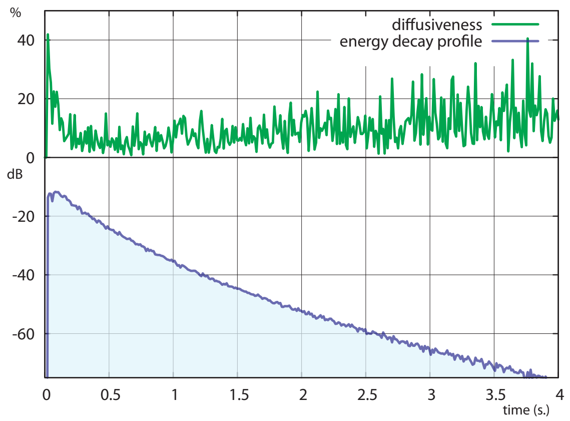

Fig. 3: Example decay profile. The top plot shows the diffusiveness index while the bottom plot shows the energy decay in one of the sub-bands. Here, we plotted the diffusiveness as the amount of *directional energy* relative to the total energy of each decay block.

and a target listening position. Our system automatically identifies identical image-sources across key locations during the pre-computation step and generates a single global list of image sources and several indexed IS sets.

An IS set is list of IS indices and reference source/listener position. It compactly encodes valid image sources for a source and listener at predefined reference locations in the environment. Several IS sets can be stored throughout an environment, for instance within each room. and directly used at run-time without having to explicitly check for visibility. We believe the IS set approach provides an efficient compromise for gaming applications, avoiding costly run-time visibility checks.

5. RUN-TIME DYNAMIC REVERBERATION

Once the image-sources (IS) locations and gradients, IS sets and decay profiles have been pre-computed for the environment, they can be queried and interpolated at run-time depending on the position of the sound source and listener to synthesize dynamic reverberation effects.

5.1. Extrapolating image sources

In a first step, the position of the source and listener is used to search through the list of IS sets and decay profiles for closest matches, first based on the current listener's position and then on the source position.

Once a combination of IS set and decay profile has been retrieved, all image sources in the set must be extrapolated to the actual source position S. Using the pre-computed gradient matrix H, each image-source position I can be obtained as:

$$I = I_0 + HS, \tag{3}$$

where S is the current source position and I_0 is the image-source location computed at the nearest keypoint and stored relative to $(0,0,0)$ (see Section 4). Extrapolating image sources leads to smooth transitions when switching between sparse key locations in the environment.

Finally, each image source and decay block will contribute an additional "virtual source" to the audio processing pipeline.

5.2. Trading image-sources for decay blocks

Rendering each individual image source requires synthesizing a delayed and filtered copy of the original signal. Depending on the chosen implementation, this may lead to a significant computational cost. A possible level-of-detail approach is to group several image sources into an equivalent decay block at run-time by integrating their energy and computing a representative direction (see Section 4). The number of individually rendered image sources can then be dynamically chosen at run-time and the remaining ones converted to an equivalent decay block and efficiently rendered as part of the reverberation model.

6. SCALABLE PROCESSING PIPELINE

In this section, we describe a scalable processing pipeline which can efficiently render large numbers of image sources and decay profiles. Our run-time engine processes and mixes concurrent blocks of input audio data, typically tens of thousands. Each block corresponds to either the direct sound from a source, an image source or a reverberation decay block. For instance, playing-back a sound source with 10 image sources and a 100-block reverberation filter queues up 111 blocks of input audio data into the processing pipeline.

6.1. Efficient prioritization and masking

Similar to [25], our pipeline is able to process a massive number of concurrent blocks by performing lossy processing in frequency domain. We first use a conservative threshold to immediately cull empty or very low energy blocks, typically below -120 dB. We then account for auditory masking between all the blocks to process. Masking between all concurrent blocks is evaluated at each frame and only audible blocks are sent down the pipeline. Additionally, the engine dynamically allocates processing to blocks contributing more energy to the final mix according to a user-specified computational budget. Each block is allocated a number of Fourier coefficients to process so that a global number of operations is enforced. For addtional details, we refer the reader to [25].

As the number of blocks to process increases, sorting and prioritizing the blocks can become a bottleneck of the approach. To increase the throughput of the pipeline, we propose to use a two-stage masking. In the first step, all the blocks are efficiently sorted by decreasing energy, based only on a broadband energy level. A conservative masking evaluation is performed using the algorithm of [25]. All audible blocks can then be routed

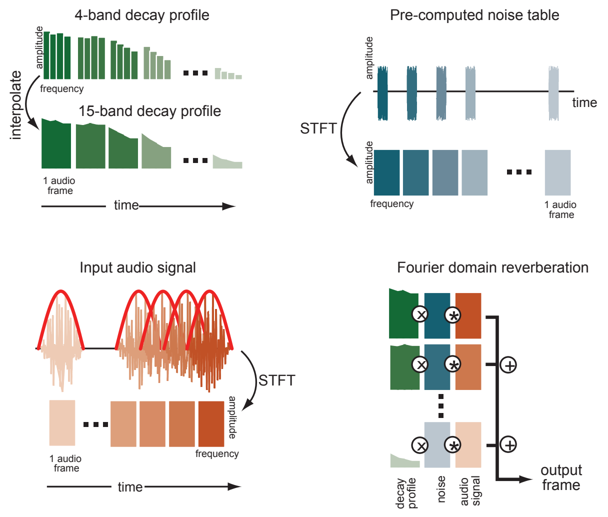

Fig. 4: Illustration of our block-based, Fourier-domain, artificial reverberator. A noise sequence is first weighted by the real-valued reverberation decay profile before its complex multiplication with the input source signal. Several past blocks of the input audio signal undergo this process and add-up to construct a frame of reverberated audio signal.

to an optional finer-grain masking stage using band levels and more conservative thresholds. Finally, all blocks that have not been discarded after the two masking stages are prioritized and prepared for processing. Since updating sub-band levels can it itself become costly for large numbers of blocks, this two-stage masking significantly increases the throughput of our pipeline. To limit runtime overhead, the raw energy present in the source signals, as well as raw sub-band levels are pre-computed and cached with their Fourier-domain data.

6.2. Frequency-domain processing

To optimize processing, all the input sounds are prestored in the Fourier frequency domain so that only a single inverse Fast Fourier Transform (FFT) per output channel is required to reconstruct an output audio frame. We typically process 1024-sample blocks at 44.1 KHz sampling-rate and use 50% overlap. Hence, our audio processing loop runs at $\Delta T = 86$Hz outputing frames of 512 reconstructed samples. For efficient data access, the same ΔT is used for both audio processing and sampling of the decay profiles. We use overlap-add reconstruction which introduces time-domain smoothing and eliminates possible artefacts due to staircase decay envelopes or when switching between IS sets.

Image sources, as well as directional components of the reverberation blocks, are rendered as delayed copies of the original signal at time T. Given the time delay t for a given image source, we first locate the closest integer input block of audio data at $T - t/\Delta T$ and apply a complex shift for the remaining fractional delay. While this is not as accurate as a time-domain implementation, we believe it is a good tradeoff for gaming applications. Additional panning or binaural filtering is used to render the position of the image-source.

To synthesize the late reverberation filter, we use a circular array of successive white-noise blocks, which we also pre-transform into the Fourier domain. To render spatially diffuse decay blocks, we generate uncorrelated noise sequences by picking a random starting block in the array for each output channel. Each block of noise is equalized in the Fourier domain using the values stored in the decay block and complex-multiplied with a block of source signal. Figure 4illustrates our reverberation process. Although we do not use zero-padding on the source signal (thus performing circular convolution), we found that the use of a Hann window limits audible aliasing.

To model frequency-dependent attenuation of image-sources and decay blocks, we use a 15-band equalization using non-uniform sub-bands. The 15 obtained gains are directly used to multiply the complex Fourier coefficients within each band. To limit the size of our data structures, we interpolate the equalization gains from only 4 frequency-dependent values stored in the image sources and decay blocks (see Section 4).

6.3. Alternative processing domains

Although our pipeline is efficient, other alternative processing domains can also be considered. For instance, the Modified Discrete Cosine Transform (MDCT) domain is widely used in audio coding and is a good alternative to render diffuse artificial reverberation effects with a quality comparable to our Fourier-based approach. Instead of a complex multiplication, only a scalar multiplication of the source signal with a noise signal is required, making it more efficient. We observed up to a $\times 6$ speed-up on an Intel Core2 Extreme 3GHz processor, with an average processing time of 0.2 msec. per audio frame for a 200-block long reverberation filter (≈ 2 sec. reverberation time) and using all frequency coefficients.

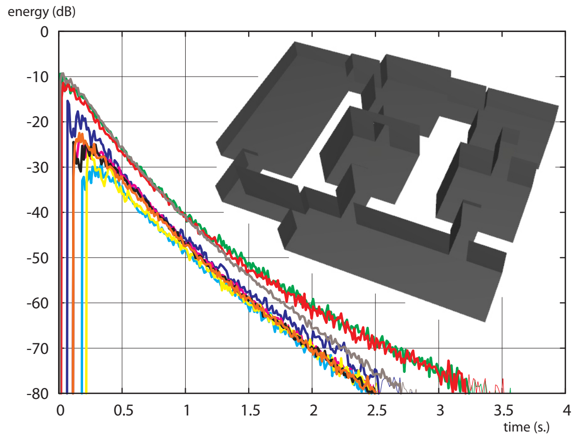

Fig. 5: Example decay profiles computed for different pairs of key locations in the pictured environment.

However, MDCT domain cannot be used for general filtering, e.g. rendering delays. As a result, image-sources cannot be efficiently rendered. A hybrid time-frequency domain approach might prove a good alternative. In that case, image sources could be rendered in time-domain with higher quality processing of delays while diffuse reverberation could be rendered in MDCT domain. The level-of-detail approach proposed in Section 5.2 could be particularly useful in that context.

7. PERFORMANCE AND RESULTS

7.1. Pre-computation

Pre-computation time typically ranges from a few minutes to a few hours depending on the complexity of the environments and number of rays used. Our ray-tracer uses a regular grid to speed-up ray-polygon intersections but was not optimized further. For 100000 rays, we observed pre-computing times ranging from 18 sec. to 35 sec. for a given source and listener pair and for a maximum propagation order of 100 and 200 respectively. In our current implementation, we trace rays for every source/listener pair, which results in pre-computation times growing quadratically with the number of key locations. However, it is possible to consider only a limited set of key locations for which a full ray-tracing is done and efficiently recompute additional image-source sets for additional locations. For instance, a conservative ray-tracing step can be amortized over entire box-shaped or spherical areas [4]. For a given source/listener position, a valid image-source set can then be quickly determined by going through the identified conservative list of potentially valid image sources. Other GA approaches, such as beam tracing, would also be more efficient albeit not as general.

For the simple two-room environment of Figure 2 pre-computation time for three key locations was about 5 minutes. A total of 100 image sources was stored for the entire environment, requiring ≈ 5 Kbytes using the structures described in Section 4. Storing 9 different IS sets of 20 image sources required ≈ 3 Kbytes, while 9 different reverberation profiles required ≈ 10 Kbytes(without quantizing similar decay blocks).

For a 14 room environment, pre-computation time goes up to 114 minutes with one key location per room (100K

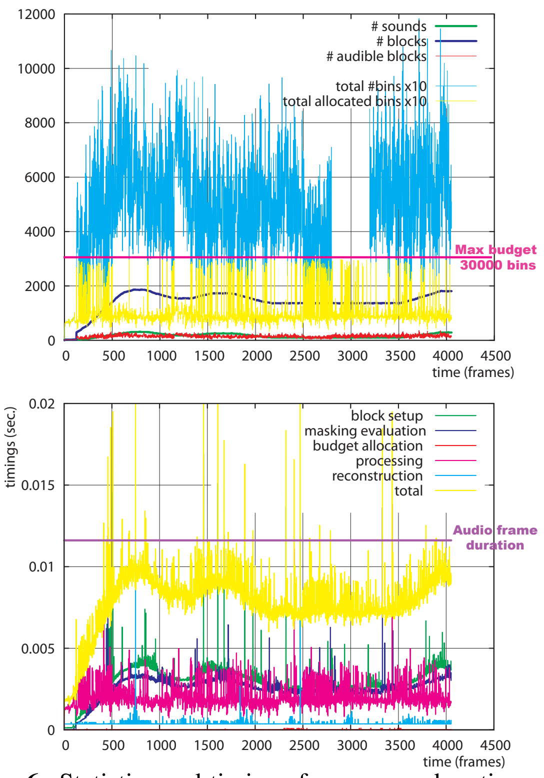

Fig. 6: Statistics and timings for our reverberation processing pipeline. In this example, the maximum number of active primary sources is about 20 and each was rendered with 10 additional image sources and a different reverberation profile. The maximum number of queued blocks is 2000 while the number of audible blocks is less than 100. The rendering budget was set to a maximum 30000 DFT coefficients per frame.

rays, 200 orders of reflection) and data structures require a total of $\approx$ 500 Kbytes (keeping 20 image sources per IS set and without vector quantization of the decay blocks). Further compression of the late decay blocks by vector quantization leads to a total size of only 300 Kbytes in that case. After the first half-second of reverberation, a large number of decay blocks can be shared between decay profiles (see Figure 5).

An alternative to computing all pairs of decay profiles for different source and receiver locations is to combine decay profiles at run-time using convolution to simulate coupling. This also allows for including additional run-time effects, such as opening or closing doors. Since decay profiles contain few samples, convolutions can be efficiently implemented. By traversing a spatial adjacency graph (e.g., cell/portal graph [24, 13, 7]) between the different spaces, significant indirect coupling routes can be efficiently identified and rendered. For additional details, we refer the reader to [23].

7.2. Run-time engine

The majority of the run-time operations are used for the actual audio signal processing. Interpolating the image sources is very efficient, requiring only three dot products per image source. It is also possible to account for source directivity at the expense of computing an additional orientation vector, which can be achieved at the expense of three additional dot products.

Our current signal processing pipeline can handle tens of thousands concurrent blocks even on modest computing platforms. Figure 6 shows statistics and timings gathered during an interactive session on a Pentium M 1.7 GHz laptop. A widely-used approach to limit processing time for late reverberation is to use a dedicated submix bus for all sources sharing the same late reverberation profile. In our engine, however, we are able to maintain a significant amount of individual source processing. In particular, all image sources and early decay blocks are re-rendered for each source leading to more dynamic and compelling reverberation effects.

Preliminary listening tests suggest that including image sources results in more convincing distance perception. Image sources also carry significant cues for surface proximity effects, resulting in increased gain and bass level as sources move closer to walls or corners.

8. CONCLUSIONS AND PERSPECTIVES

We proposed an approach to sample and pre-compute parametric reverberation effects for interactive applications. Our approach performs costly geometrical calculations off-line and does not require explicit storage or access to the geometry at run-time. Instead, we proposed compact data structures and solutions to interpolate image-sources and late reverberation as the sources and listener move. In the future, we would like to further explore how reverberation effects in 3D environments can be automatically and efficiently sampled to avoid specifying key locations by hand. We also introduced several improvements to a scalable frequency domain pipeline supporting directional and diffuse spatial processing. This work suggests that a tighter integration of audio coding and processing could be extremely beneficial for interactive applications, such as games.

9. REFERENCES

[1] Durand R. Begault. *3D Sound for Virtual Reality and Multimedia.* Academic Press Professional, 1994.

[2] B.-I. Dalenbäck. CATT Acoustic. http://www.catt.se/.

[3] Bengt-Inge L. Dalenbäck. Room acoustic prediction based on a unified treatment of diffuse and specular reflection. *J. Acoustical Soc. Am.*, 100(2):899–909, August 1996.

[4] Thomas Funkhouser, Patrick Min, and Ingrid Carlbom. Real-time acoustic modeling for distributed virtual environments. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 365–374, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[5] Thomas Funkhouser, Nicolas Tsingos, Ingrid Carlbom, Gary Elko, Mohan Sondhi, James E. West, Gopal Pingali, Patrick Min, and Addy Ngan. A beam tracing method for interactive architectural acoustics. *J. Acoustical Soc. Am.*, 115(2):739–756, 2004.

[6] W. Gardner. *Reverberation Algorithms, Applications of Digital Signal Processing to Audio and Acoustics, Chapter 6.* Mark Kahrs and Karlheinz Brandenburg Ed., Kluwer Academic Publishers, 1998.

[7] Denis Haumont, Olivier Debeir, and François Sillion. Volumetric cell-and-portal generation. In

Computer Graphics Forum, volume 3-22 of *EUROGRAPHICS Conference Proceedings*. Blackwell Publishers, 2003.

[8] R. Heinz. Binaural room simulation based on an image source model with addition of statistical methods to include the diffuse sound scattering of walls and to predict the reverberant tail. *Applied Acoustics*, 38:145–159, 1993.

[9] Jean-Marc Jot. Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *Multimedia Systems*, 7(1):55–69, 1999.

[10] K.H. Kuttruff. Auralization of impulse responses modeled on the basis of ray-tracing results. *J. Audio Eng. Soc.*, 41(11):876–880, November 1993.

[11] Samuli Laine, Samuel Siltanen, Tapio Lokki, and Lauri Savioja. Accelerated beam tracing algorithm. *Applied Acoustics (to appear)*, 2008. Available online at http://dx.doi.org/10.1016/j.apacoust.2007.11.011.

[12] C. Lauterbach, A. Chandak, and D. Manocha. Interactive sound rendering in complex and dynamic scenes using frustum tracing. *Transactions on Visualization and Computer Graphics*, 13(6):1672–1679, Nov.-Dec. 2007.

[13] Sylvain Lefebvre and Samuel Hornus. Automatic cell-and-portal decomposition. Technical Report 4898, INRIA, July 2003.

[14] Tobias Lentz, Dirk Schröder, Michael Vorländer, and Ingo Assenmacher. Virtual reality system with integrated sound field simulation and reproduction. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 70540, 19 pages, 2007. doi:10.1155/2007/70540.

[15] J. Merimaa and V. Pullki. Spatial impulse response rendering. *Proc. of the 7th Intl. Conf. on Digital Audio Effects (DAFX'04), Naples, Italy*, October 2004.

[16] J.M. Naylor. Odeon - another hybrid room acoustical model. *Applied Acoustics*, 38(1):131–143, 1993.

[17] Jackson Pope, David Creasey, and Alan Chalmers. Realtime room acoustics using ambisonics. In *The Proceedings of the AES 16th Intl. Conf. on Spatial Sound Reproduction*, pages 427–435. Audio Engineering Society, April 1999.

[18] D. Rocchesso. *Spatial Effects, DAFX - Digital Audio Effects, Chapter 6*. Udo Zölzer Ed., Wiley, 2002.

[19] Dirk Schröder, Philipp Dross, and Michael Vorländer. A fast reverberation estimator for virtual environments. In *Proceedings of the AES 30th Intl. Conf., Saariselkä, Finland*, March 2007.

[20] M.R. Schroeder. Natural sounding artificial reverberation. *J. Audio Eng. Soc.*, 10(3):219–223, 1962.

[21] Samuel Siltanen, Tapio Lokki, Sami Kiminki, and Lauri Savioja. The room acoustic rendering equation. *J. Acoustical Soc. Am.*, 122(3):1624–1635, September 2007.

[22] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, New York, NY, USA, 2002. ACM.

[23] E. Stavrakis, N. Tsingos, and P. Calamia. Topological sound propagation with reverberation graphs. *Acta Acoustica United with Acustica, Special issue on Virtual Acoustics*, (1), 2008.

[24] Seth Teller. *Visibility Computations in Densely Occuded Polyhedral Environments*. PhD thesis, Computer Science Div., University of California, Berkeley, 1992.

[25] Nicolas Tsingos. Scalable perceptual mixing and filtering of audio signals using an augmented spectral representation. In *Proceedings of the Intl. Conf. on Digital Audio Effects*, September 2005. Madrid, Spain.

[26] Earl Vickers, Praveen G. Krishnan, and Ravirala N.K. Sadanandam. Frequency domain artificial reverberation using spectral magnitude decay. In *Proceedings of the 121th AES convention, preprint 6926*, Oct 2006.

[27] M. Vorländer. Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm. *J. Acoustical Soc. Am.*, 86(1):172–178, 1989.