Eurographics Symposium on Parallel Graphics and Visualization (2014)
M. Amor and M. Hadwiger (Editors)

Precomputing Sound Scattering for Structured Surfaces

G. Mückl[1] and C. Dachsbacher[1]

[1]Karlsruhe Institute of Technology

Abstract

Room acoustic simulations commonly use simple models for sound scattering on surfaces in the scene. However, the continuing increase of available parallel computing power makes it possible to apply more sophisticated models. We present a method to precompute the distribution of the reflected sound off a structured surface described by a height map and normal map using the Kirchhoff approximation. Our precomputation and interpolation scheme, based on representing the reflected pressure with von-Mises-Fisher functions, is able to retain many directional and spectral features of the reflected pressure while keeping the computational and storage requirements low. We discuss our model and demonstrate applications of our precomputed functions in acoustic ray tracing and a novel interactive method suitable for applications such as architectural walk-throughs and video games.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing; H.5.5 [Information Systems]: Sound and Music Computing—Signal analysis, synthesis, and processing

1. Introduction

Computer simulations of room acoustics have made steady progress since their early beginnings. Over time, the methods for room acoustics developed striking similarities to those used for computing global illumination in computer graphics. In fact, it is safe to say that researchers in both fields have looked at each other's work for inspiration. Yet, despite many similarities in the simulation methods for room acoustics and global illumination, the former have to deal with additional challenges like the time dependence of sound propagation. Thus, many new ideas and methods have been applied to global illumination problems but have not yet been successfully adapted to room acoustics. One apparent instance is the handling of wall reflections. Most simulations use a simplified model in which the incoming sound is reflected partially perfectly specular and partially perfectly diffuse. A simple scattering coefficient is used to determine the ratio between the two. This model was already used in the early acoustic ray tracing simulations [KSS68] and has become the most dominant one, although it is well known that wall reflections on rough walls behave in a more complex way [DKS93]. Such more complex behaviours have been studied numerically and experimentally [CD09]. Still, there were few attempts to incorporate these results in models used for computer simulations. Often, the properties of a structured or rough surface are reduced to scattering and absorption coefficients. We are aware of only one work using a more sophisticated analytical BRDF for room acoustics, which was derived for random rough surfaces [DES99].

This paper presents a method for numerically computing acoustic BRDFs for surface structures defined by height and normal maps. Our work shares ideas with the predicted reflectance functions presented by Westin et al. [WAT92]. We compute the BRDFs for single scattering using the Kirchhoff approximation based on a method presented by Tsingos et al. [TDLD07]. We can compute either the near field reflections at various distances or restrict ourselves to the far field where we can compute a BRDF that can be used for acoustic ray tracing. Precomputation is challenging in both time and storage space consumption as the reflected pressure is in the most general case a 7d function. We show how to significantly reduce the required effort by a careful choice of sampling points in this parameter space combined with an appropriate interpolation method. Finally, we demonstrate acoustic ray tracing and computing single scattering in virtual scenes on the GPU as applications.

2. Related Work

Acoustic ray tracing was derived during the 1960s from a Monte-Carlo method for determining the mean free path in a room [AN58a, AN58b]. It was extended to tracing full paths and applied to various problems early on (e.g. [KSS68, Sch70]). These early models computed mirror reflections and sometimes generated random new ray directions in accordance with Lambert's law. The continued development of the method focused on variance reduction (among others [Lam96, DKS93]) and eventually spawned techniques like cone tracing [Dal96] and image sources [AB79]. However,

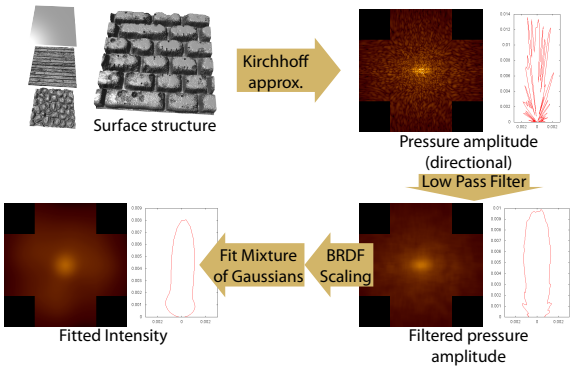


Figure 1: *Schematic overview over the precomputation method: First, the pressure amplitude is computed from a height map, then filtered and fitted. A rescaling step is added for BRDF precomputation. The data shown is from a 1m×1m patch. Source and receiver distances are 10m. The source is directly above the patch centre and emits at 10kHz.*

the modelling of wall reflections still received little attention. Embrechts et al. [DES99] derived a BRDF analytically from the Kirchhoff approximation for random rough surfaces and used it in a phonon tracing method. He showed that the Kirchhoff approximation and acoustic ray tracing do produce the same results in the limit of negligibly small wavelengths [Emb00b]. Also, he estimated boundary conditions for the validity of using BRDFs in ray tracing [Emb00a].

The Kirchhoff approximation is often used to compute scattering for a given surface geometry and obtain either a directional scattering or a single direction-independent scattering coefficient [CD09, EAS01]. Similarly, measurements of directional scattering properties are often reduced to a single scattering coefficient as well [CDD*06]. Tsingos et al. [TDLD07] presented a method to compute real-time single scattering off of arbitrarily shaped highly detailed surfaces using the Kirchhoff approximation. Our method for computing the scattered pressure in the precomputation is based on the formulation of the Kirchhoff approximation given in this paper. The precomputation of diffuse and glossy reflection is well established in computer graphics [Ram09], but cannot be applied to acoustics in a straight-forward manner because wave-based phenomena are not included.

Siltanen et al. derived the acoustic analogue of the rendering equation [SLKS07]. It differs from Kajiya's equation for light propagation [Kaj86] only in that it includes a phase shifting operator based on the travelling time of the sound between surfaces. Due to the similarity between the two equations it is possible to adapt many algorithms for solving global light transport for room acoustics.

3. Precomputation and Evaluation of Acoustic BRDFs

Many models of varying computational complexity and accuracy are available for computing the sound scattered off a structured surface. Our method needs to cover all combinations of incoming and outgoing directions as well as the distance to the sound source and receiver. Thus, computational complexity is an issue and we selected the Kirchhoff approximation as the basis for its simplicity. Other models like the Boundary Element Method and its variations are more accurate, but also much more expensive and therefore less practical. Our model is applicable to input generated by these more sophisticated methods with only minor changes.

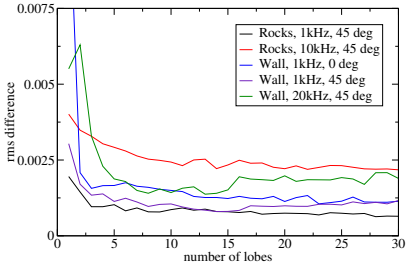


Figure 2: *Root mean square difference between fitted function and originally computed function values for varying numbers of vMF lobes. Note that a noticeable reduction in error takes place only for the first few lobes.*

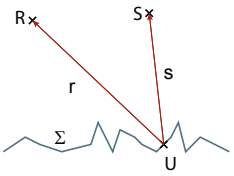
3.1. Kirchhoff Integral

Tsingos et al. [TDLD07] derive the following form of the Kirchhoff approximation for computing the single scattering of a structured surface Σ:

$$\begin{aligned} P(R) &= P_0(R) + \int_\Sigma \hat{p}(R)dS \\ &= P_0(R) + \int_\Sigma (p(R) + \rho(U)p'(R))dS, \text{ with} \end{aligned} \quad (1)$$

$$p(R) = -\frac{P_0 e^{ik(s+r)}}{4\pi\rho r}\left(\left(ik - \frac{1}{s}\right)(\vec{ns}) + \left(ik - \frac{1}{r}\right)(\vec{nr})\right)$$

$$p'(R) = -\frac{P_0 e^{ik(s+r)}}{4\pi\rho r}\left(\left(ik - \frac{1}{s}\right)(-\vec{ns}) + \left(ik - \frac{1}{r}\right)(\vec{nr})\right),$$



where k is the wave number, $\rho(U)$ the surface reflectivity at the surface point U; s is the distance between the source and U; r is the distance from U to the receiver at R, $\vec{n}$ the surface normal at U, $\vec{s}$ a unit vector pointing from U to the source at S and $\vec{r}$ a unit vector pointing from U to R. Thus, this integral approximates the frequency dependent pressure amplitude for the sound scattered back from the surface to a specific receiver position. Multiple scattering is neglected.

3.2. Precomputation

The integral given above is a function of seven parameters: the relative positions of source and receiver to the surface, and the frequency. Given a structure of a rigid surface encoded in a normal map and corresponding height map, the reflection behaviour of that surface can be computed. To avoid computation of the whole integral for each surface reflection in a room acoustics simulation, we precompute and store the function results instead. However, a sufficiently

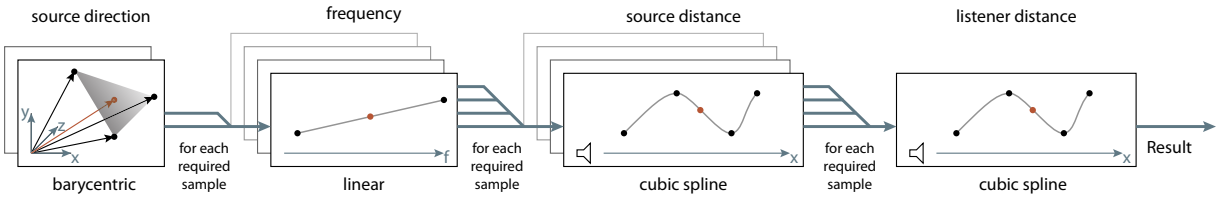


Figure 3: *Interpolating between the precomputed functions is a process where each step supplies the supporting values for the following one: the barycentric interpolation for source direction is performed once for each support required by the linear frequency interpolation, which in turn is performed once for each support for the source interpolation and so on.*

dense sampling and lossless storage of the full results is expensive in both computational time and storage space. Thus, we propose the following practical sampling and interpolation scheme for these *reflected pressure functions* (RPF).

First, we compute the complex reflection for listener directions parameterised using a hemicube map for a fixed listener distance, source position and frequency. The resulting function shows very high frequency variations in the listener direction dependence in both amplitude and phase of the reflected pressure. Keeping the full information for these maps is not practical as the resulting amount of data would become too large to be used efficiently in simulations. Therefore we only keep the amplitude and discard phase information. We apply a low pass filter to the amplitude map to extract the most prominent reflection directions. Next, we fit von-Mises-Fisher (vMF) functions (mixture of Gaussians) via an expectation maximisation (EM) algorithm that uses a point cloud (typically 2000 samples) generated by rejection sampling of the hemicube map [HSRG07]. Since the EM algorithm alone does not scale the relative intensities of the individual Gaussians we rescale them to match the hemicube values at the respective centre directions of each lobe. Fig. 1 shows plots of typical results at various stages of the process.

With this approach we can represent the original function with a number n of Gaussian lobes parameterised by a central direction, width and scaling factor. That is, $4n$ floating point values are sufficient to represent a good approximation of the amplitude of the reflection for a given listener distance, source position and frequency. We found 5 lobes to be sufficient to cover the most important reflection directions for the range of surface structures that we tested (see Fig. 1). A higher number of lobes did not result in a significantly better convergence (see Fig. 2). The low pass filter removes higher frequency details and thus, the convergence to the original function is limited.

This integration and fitting process is repeated for different frequencies, source directions, source distances, and receiver distances to sample a full acoustic response of a structured surface. The reflected pressure field generally varies most when close to the surface and becomes much smoother with increasing distance. Thus, we distribute the sampling points for the source and listener distances on a cubic scale between the minimum and maximum distances to get a much denser sampling closer to the surface. The maximum source and listener distances should be chosen in such a way that the reflected pressure decreases with $\frac{1}{r}$ beyond this distance so that a faithful extrapolation is possible. In our tests, we empirically determined this distance by computing the distance dependence for a set of sample directions and manually choosing an appropriate cut-off value.

Unless otherwise noted we computed our RPFs with the following parameters: square patches with each edge 1m long, every other frequency of the Bark scale (12 frequencies in total between 50Hz and 10.5kHz [Zwi61]), 15 distances for both source and receiver distributed between 1m and 20m with distances between sample points increasing cubically, 160 equally distributed random directions for the source position and a hemicube map resolution of 40×40 for the full side and 20×40 for the four halved sides. Using 5 lobes, this results in $9\cdot10^7$ float values that need to be stored for a complete RPF. We chose these parameters with typical rooms and small halls in mind.

These precomputed RPFs are only accurate when the geometry of the patch and the irradiation conditions stay the same. If parts of the surface are cut off or occluded, the integration area changes and the shape of the RPF changes globally. Similarly, enlarging surfaces adds contributions not present in the precomputed RPF. Yet, we neglect these changes as precomputing RPFs for each patch shape in a scene is too expensive in practice. Instead, we assume that the RPF for a sufficiently sized regular patch approximates occluded or oddly shaped patches well enough. These restrictions highlight the fact that our RPFs describe the reflection behaviour of individual surface tiles rather than complete surfaces. They are a well-suited intermediate between explicitly modelled large-scale geometry and small-scale structures requiring explicit dense sampling. But it is possible to create RPFs for patch area ranges, e.g. by tiling the input and using the closest RPF for the actual surface size.

3.3. Interpolation

In order to reproduce the reflected pressure for arbitrary source and receiver positions and frequency from our precomputed data we interpolate between the values of the vMF functions for the nearest available parameter values. Our algorithm for this is a nested chain of interpolation steps, where each step provides the source values for the sampling

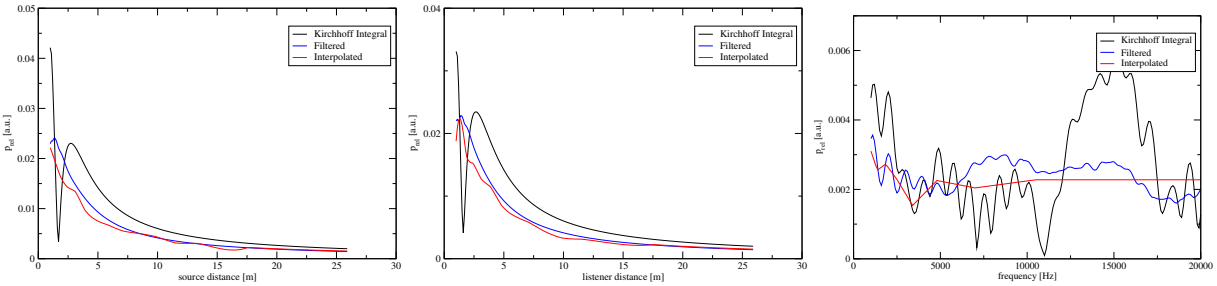


Figure 4: *Interpolation accuracy: these plots show computed and interpolated results for the wall surface for (in order) source, listener and frequency interpolation. The values for the non-interpolated parameters were chosen to correspond with sampling points for the precomputation: source and listener distances are the biggest distance sampling point at 17.6m, the frequency was 4.8kHz, the source and listener direction correspond to the source direction closest to the normal direction.*

points required for the next one. From the innermost to the outermost level, the individual steps are (see also Fig. 3):

- source direction interpolation between the 3 closest sampled source directions using barycentric coordinates
- linear interpolation of the closest sampled frequencies
- local cubic spline interpolation for the source distance using a spline segment through the nearest four sampled source distances
- cubic spline interpolation for the listener distance

Thus, $3 \cdot 2 \cdot 4 \cdot 4 = 96$ individual vMF functions are evaluated for one interpolated value. The search for the three required nearest source directions for barycentric interpolation is accelerated using a cubemap as a lookup table. The map stores at each point the values of the three nearest sampled directions and the barycentric coordinates. For frequencies outside the sampled range only the closest frequency is used and no extrapolation takes place. Similarly, there is no extrapolation for distances smaller than the smallest sampled distance. For larger distances, the reflected pressure is extrapolated with a $\frac{1}{r}$ function.

Fig. 4 shows a typical comparison of interpolated results to the original computed and filtered values. Clearly, the filtering introduces deviations from the original values. These are most noticeable in the frequency dependence. This can be explained by the direction of the narrow peaks in the unfiltered reflected pressure which rotate towards the surface normal with increasing frequency. This fine structure is responsible for the frequency dependence visible in the unfiltered data, but is not retained by our low pass filter, resulting in the observed smoothing. However, the interpolated values match the filtered values quite closely in every case. This interpolation method provides good results for the range of surface structures that we tested.

3.4. Absorbing surfaces and surfaces with holes

Our method can be trivially extended to partially and inhomogeneously reflecting surfaces by adding an additional map for the complex reflection factor instead of assuming $\rho(U) \equiv 1$. Also, this model is not limited to reflections.

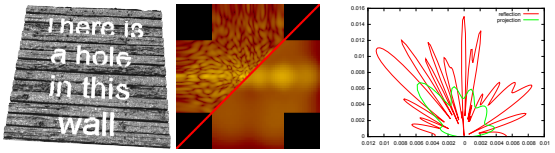


Figure 5: *Diffraction of sound by a surface with holes using our precomputation scheme: From left to right: input, hemicube map of the absolute values of the integral, fitted values, and a horizontal slice through the maps as polar plot. The diffracted sound is distributed similarly to the reflected sound and is processed in the same way. The data shown is for sender and receiver distances of 10m and 10 vMF lobes.*

The Kirchhoff integral in its original form can be used to compute diffraction caused by sound waves reflected by or travelling through holes in a screen [FHLB99]:

$$P(R) = P_0(R) + \int_H \frac{P_0 e^{ik(s-r)}}{4\pi rs} \left(\vec{nr} \left(-ik - \frac{1}{r} \right) - \vec{ns} \left(ik - \frac{1}{s} \right) \right) dS \,. \tag{2}$$

H denotes the surface without the hole. We adapted our method to precompute the diffraction caused by holes. Fig. 5 shows a typical result for such a diffraction pattern with listener positons on the other side of the surface. These results were obtained by computing the integral without the contribution of the direct sound. But the widened distribution of the diffracted sound pressure requires a higher number of lobes to obtain a proper fit. In our example we used 10 lobes.

3.5. Discussion of Approximations

Our method depends on several approximations, namely discarding phase information, low pass filtering of the pressure amplitude and the suitability of vMF functions for fitting.

We discard phase information for several reasons. First, keeping the phase information is a practical issue as it would add a lot of data. It has a fine structure comparable to that of the amplitude (see Fig. 6). Fitting that data directly using, for example, vMF functions, will lead to a representation that is not significantly smaller than the raw data. Applying a lossy data reduction scheme like the low pass filtering we

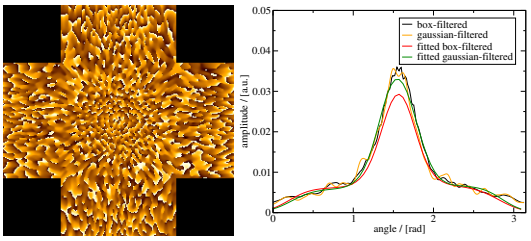


Figure 6: *Left: exemplary result for the directional phase shift. We know no analytical function with a reasonably small set of coefficients which can be fitted to such data. Right: influence of the filter on the result: central cross-sections of the hemicube maps for filtered and fitted amplitudes when applying a box- or Gauss-shaped low pass filter. The choice of the filter kernel has little effect on the result. The data shown is for the wall surface with sender and receiver distances of 5m. The sender is centered over the patch.*

use for the amplitude is not possible, either. As interference effects have a non-linear dependency on the phases of the contributing waves, inaccurate modelling of phase shifts on wall reflections can change the results drastically. Second, it is not always required to simulate phase and interference for accurate room acoustics modeling. We discuss this in more detail in section 4.

The low pass filtering is performed to remove details from the angular distribution of the pressure amplitude which we assume to be inaudible in a realistic setting. Consider the sound reflected off a tiled wall where each tile has a finely patterned directional reflectivity. The incoming sound waves hit each tile at slightly different angles, causing the outgoing sound to move into slightly different directions. It is easy to see that these slightly different intensities for each tile average out at the location of the receiver in such a way that high-frequency details in the reflection behaviour get "blurred out". We chose to use a box filter as low pass filter. Using different filters did not have a noticeable impact on the final results of our precomputation. After the filtering step, the amplitude data typically shows a couple of lobes that can be represented by von-Mises-Fisher functions with acceptable errors (see Fig. 6).

4. Application to single scattering

The precomputed RPFs can be applied to the computation of single scattering. For this, we modified *Instant Sound Scattering* (ISS) from Tsingos et al. [TDLD07]. This GPU-based real-time method computes single scattering in a dynamic scene in screen space. First, this method renders a map of the scene from the point of the sound source containing the surface location and normals. Then, for each frequency in the spectrum the integrand of the Kirchhoff approximation is computed for every pixel and summed up.

Using precomputed RPFs, we can modify this method to avoid the per-frequency screen space integration and the related undersampling of geometry seen at grazing angles. Also, like ISS, our method's performance is decoupled from the complexity of the input geometry. It groups visible surfaces by material, position and surface orientation and then computes the contribution to the overall scattering for each group using RPFs. First, a fragment shader assigns group IDs to the visible surfaces while rendering the scene into an ID buffer with depth testing enabled so that only visible surfaces are considered. It computes a group ID from the material index, the fragment position within a uniform grid in normalised device coordinates, the x and y components of the view space surface normal and the tangent orientation. The tangent orientation can be stored as a single value. The normal defines the plane of the tangent and the orientation within that plane is determined by a single angle. We obtain the reference axis for this angle by forming a vector orthogonal to the normal from the x axis direction. Thus, each group ID identifies a unique set of materials, positions and orientations for which the RPF has approximately the same value.

We then generate a histogram of group IDs. For the non-zero entries in the histogram, the corresponding RPF contributes to the final spectrum. Similarly to the original ISS, the contribution needs to be scaled with the estimated surface area derived from the number of fragments in the histogram bin. The number of group IDs occurring in a frame is bounded by the framebuffer resolution in the first step and is considerably lower than the number of pixels required in ISS for achieving comparable quality (see also section 6.3). Finally, we auralize the resulting sound from the generated spectrum using the method described in [TDLD07].

To prevent the histogram from becoming too large, the parameters for the discretisation must be chosen carefully. For a single material we find it sufficient to use 16 subdivisions on each axis for each of the screen space coordinates, normal components and 32 subdivisions for tangent orientations. Thus, 2^{25} histogram bins, requiring 128MB of memory, are needed for each material in addition to the memory for the RPFs themselves.

Since our RPFs do not have phase information, our results differ significantly from those obtained by ISS as the interference effects at the listener position are missed (see accompanying video). Whether neglecting phase shifts is appropriate or not depends on the coherence time of the sound that is ultimately auralized. The coherence time in this context is the time span during which the source signal keeps a constant phase. A single sinus tone has a coherence time as long as the tone itself, while white noise has a negligible coherence because it will, due to its nature, constantly jump between phase angles without correlation. Interference effects are only observable when the travelling time difference between paths is smaller than the coherence time. If the time difference is larger, no stable interference exists: the parts of the signal that arrive at the listener at the same time do not have a constant phase relation between them anymore and are uncorrelated. While artificial sounds or tones from indi-

vidual musical instruments can have long coherence times and thus observable interference effects in reflected sound, more complex sounds (e.g. music or voice) often do not. Thus, we assume our method to be more suitable for computing the scattering of complex sounds than ISS.

5. Application to multiple scattering using ray tracing

While RPFs are readily applicable to the simulation of single scattering, the precomputation can be modified to obtain more general bidirectional reflectance distribution functions (BRDF). These functions are completely analogous to BRDFs as they are known in computer graphics and describe the directional distribution of the reflection of the radiance arriving at the surface at a specific angle. Equivalently to optics, an acoustic BRDF is defined as the variation of the outgoing radiance dL in a direction ω_o based on the change of the incoming irradiance dE from a direction ω_o [SLKS07]:

$$f(\omega_o, \omega_i, \mathbf{x}) = \frac{dL_o(\omega_o, \mathbf{x})}{dE_i(\omega_i, \mathbf{x})} \approx \frac{\Delta L_o(\omega_o, \mathbf{x})}{\Delta E_i(\omega_i, \mathbf{x})}, \qquad (3)$$

where ΔE_i and ΔL_o denote the irradiance and radiance relative to small finite solid angles and small areas. Note that radiometric quantities for optics and acoustics are exactly the same. Assuming that a patch with area A around $\mathbf{x}$ is irradiated by a source far away, we can express the irradiance as $\Delta E_i = \Phi_i / (\Delta A \cos \omega_i)$ with Φ being the power of the incoming sound. Φ_i is related to the outgoing power Φ_o by the total reflectance of the patch: $\Phi_o = R_{\text{tot}} \Phi_i$. The outgoing radiance can be expressed as radiant exitance $\Delta L_o = \Delta E_o / \Delta \omega_o$. Thus, we can express the BRDF as

$$f(\omega_o, \omega_i, \mathbf{x}) = \frac{R_{\text{tot}} A \cos(\omega_i) \Delta E_o(\omega_o, \omega_i, \mathbf{x})}{\Phi_o \Delta \omega_o}. \qquad (4)$$

The radiant exitance can be computed from the result of the Kirchhoff integral as $\Delta E_o = p_o^2 / \rho c$. We identify $\Delta \omega_o$ as the average solid angle covered by the pixels in the hemicube map of p_o. This leaves the total reflectance R_{tot} as the sole remaining parameter.

We must match the setup for the Kirchhoff integral to the assumptions made in this derivation. In particular, the incoming and outgoing waves must resemble tight, parallel beams of plane waves. The Kirchhoff integral as given in Eq. 1 models an isotropic point source emitting spherical wavefronts. Thus, we approximate incident plane waves by choosing the source distance such that the maximum phase shift of the incident spherical wave between the centre and the corner of the surface is smaller than a given threshold (typically $\frac{\lambda}{4}$ for the shortest wavelength). The BRDF is precomputed only for this distance. Since it describes parallel incoming and outgoing beams with constant intensity along their path, no further distance dependence is assumed.

The precomputation method for RPFs has to be adapted only slightly: after the filtering step, we scale the pressure hemicube map such that the integral over $p_o^2 \cos \omega_i$ equals one. That is, we divide each pressure value by the square root of the integral value. This results in an energy conserving BRDF, allowing the desired total reflectance to be kept as a parameter. The vMF fitting and the remaining process need no further adaptation. During the interpolation, the distance interpolation steps are no longer required, reducing the algorithm's complexity drastically. Note that the shortcomings related to surface shape and surface size which we previously mentioned for RPFs apply to these BRDFs as well.

Frequency	Theory	BRDF	Raytracing
1400	78.52	75.6	65.5
1500	66.16	66.8	64.0
1600	59.04	59.4	56.0
1700	53.81	54.4	50.5
1800	49.66	49.8	46.0
1900	46.23	46.0	42.5
2000	43.31	42.7	40.0
2100	40.79	40.6	33.5
2200	38.58	35.5	36.0
2300	36.62	33.3	34.0

Table 1: *Comparison of angles for the second maximum of reflected intensity for a reflection grating. For the BRDF the central direction of the largest lobe not pointing back at the sender is shown. The ray tracing result was determined by moving the detector in 0.5 degree steps in the plane of the reflection and determining the angle with the highest reflected intensity. The results are in rather good agreement.*

These BRDFs can be used together with the acoustic rendering equation given by Siltanen et al. [SLKS07]. As this equation is similar to the rendering equation for light we can directly apply well-known parallel methods from computer graphics. We implemented a straight-forward Monte Carlo tracing scheme to simulate multiple scattering. Rays start from the source into random directions (akin to light tracing [DW94]). Upon an intersection with a wall surface, a new random direction is chosen and the ray's intensity is weighted by the BRDF. At each wall intersection a visibility test of the listener location is performed and the ray's intensity is registered. Due to missing Helmholtz reciprocity, adjoint BRDFs must be explicitly computed if tracing paths starting from the listener is desired (see also [Vea97]).

6. Results

Except for the computation performed on a cluster, all timings were measured on an Intel i7 980 at 3.3GHz with 12GB system RAM and a nVidia GTX 580 with 3GB VRAM.

6.1. Precomputation

The precomputation of the complete RPF was performed on a cluster of 16 nodes with Intel Xeon E5620 running at 2.4GHz, 24GB RAM and nVidia GTX 480 GPUs. The problem lends itself naturally to parallelization because the individual hemicube maps and vMF fits are computed independently. Thus, we distributed these steps across the nodes using MPI for communication. Also, we accelerated the computation of each Kirchhoff integral within the hemicube

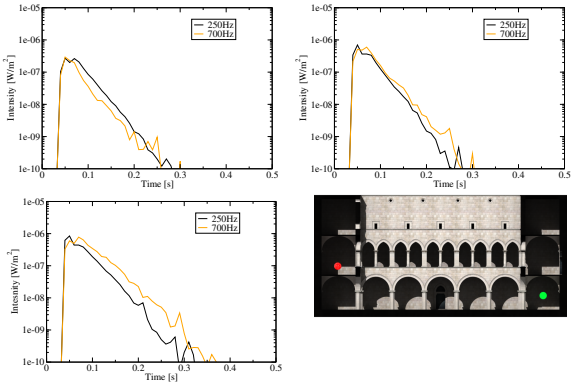
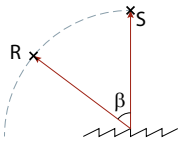


Figure 7: *Simulated reverberation for the Sponza atrium using BRDFs for three different surface structures, from top left to bottom right: a planar surface, a brick wall and a wall of smooth rocks. The variation in the results is due to the different diffusivity of these surfaces. All simulations were run with* 10^8 *rays. The source was placed in the ground level gallery, the receiver opposing it on the first floor.*

map generation step using CUDA. Similarly, the expectation maximation contains many easily parallelizable operations and is therefore also implemented using CUDA. All other steps are non-compute-intensive and were performed on the CPU. No data-intensive communication is required between the nodes as only the source position and listener distance are required for each fit and the fitting result consists only of the parameters for vMF functions. Precomputing the RPF for a surface texture of 512×512 pixels took approximately 350 minutes. Our BRDFs are much less computation-intensive and typically took about 14 minutes for the same map size on our Intel i7 980 computer.

6.2. Ray tracing



We used a sawtooth-shaped reflection grating to test the accuracy of the directivity of our BRDFs. The grating was 4m×1m with each line of the grating being 0.25m wide and arranged at an angle of 20 degrees to the normal. This grating shows sharp reflections in the direction of the second order maxima. We positioned the sound source 10m from the surface centre in normal direction and moved the listener on a semicircle with the same radius covering the expected second order reflections. As Table 1 shows, the reconstruction of the maxima locations from the computed BRDF does not vary much from the theoretical prediction of their locations. For the ray tracing result, we modeled the grating as a planar surface of the same size with a precomputed BRDF. The deviations are caused by the lack of phase changes in the BRDF and the application of the same BRDF to all rays hitting the plane regardless of the location of the hit point. This smears out the BRDF because the angles for which the BRDF is computed necessarily vary across the surface. Nonetheless, this example demonstrates that major features in surface reflections may be retained accurately in the BRDFs.

Subdivisions			fps		
Grid	Normal	Tangent	avg	min	max
16	64	16	8.33	5.78	13.86
16	64	32	7.51	5.52	11.74
16	256	16	6.61	4.35	10.90
16	256	32	5.84	3.92	9.00
32	64	16	5.60	4.42	7.63
32	64	32	4.74	3.75	6.02
32	256	16	3.70	2.78	4.56
32	256	32	2.63	2.31	3.02

Table 2: *Performance of our method using precomputed RPFs in the Sibenik scene for varying discretisation for the surface area. Since the performance varies with the viewpoint, the fps values given are the minimum, maximum and average values for a receiver moving in a straight line through the length of the scene. Examples for the results of each parameter set are provided as additional material.*

Fig. 7 shows the simulated reverberation of the Sponza Atrium with different BRDFs. The impulse response varies with diffusivity of the surface reflection, which is determined entirely by the surface structure from which the BRDF was computed. The only free parameter remaining is the average surface reflectivity, which we set to 1. Care must be take when sampling the BRDFs at high frequencies as the reflections become increasingly directional. Lower frequencies can be easily sampled even without importance sampling. We disregard edge diffraction, which occurs predominantly around the pillars. Thus, the chosen BRDF is the main influence on the result. While is not strictly valid to use a BRDF computed for a larger patch size for the many small surfaces of the ornaments, these surfaces are hit by relatively few rays. Thus, their contribution to the result is small.

6.3. Single Scattering

Our implementation renders a texture containing group IDs in OpenGL and builds a histogram for this ID texture in CUDA. One thread is launched per ID buffer pixel, which atomically increments the corresponding histogram bin. This histogram is then copied to host memory and used to estimate surface areas. RPF interpolation and the spectrum construction is done on the CPU. For comparison, we implemented the original Instant Sound Scattering using OpenGL.

For performance measurement, the histogram evaluation and RPF interpolation used twelve threads. Instant Sound Scattering achieves 37fps with 12 frequencies and a buffer resolution of 2048×2048 pixels and 11fps at 4096×4096 pixels. Resolutions less than 4096×4096 produce audible artifacts because of insufficient sampling. While the runtime of ISS depends on the number of frequencies and the size of the framebuffer used for the integration step, our method depends mostly on the number of histogram bins for which RPF interpolation is performed. We measured an average execution time of 22μs for each RPF evaluation (see Tab. 2).

The number of non-zero histogram bins increases with the variations of normals and tangents that are visible and therefore performance decreases. Even with the lowest number of bins, the results for our method do not exhibit the same level of audible artifacts and thus are qualitatively superior to ISS results rendered at 4096×4096 and comparable to those obtained at 8192×8192. Thus, our method is faster than Instant Sound Scattering for comparable quality.

7. Conclusion and Outlook

Previous definitions of acoustic BRDFs were focused on analytic derivations of the surface scattering occurring in special cases like randomly rough or sine shaped surfaces. We have shown that it is possible and practical to numerically precompute and use BRDF models derived from explicitly defined surface geometry. Our method discards phase information and much information about the angular distribution of the reflection. Yet it produces a representation of the BRDF that retains major features accurately. We also presented two applications for these precomputed functions. Our BRDFs allow us to replace the scattering coefficient parameter in acoustic ray tracing with a more detailed model. Using RPFs, our single scattering simulation method becomes less dependent on the frame buffer resolution. At almost interactive rates, it is faster than Instant Sound Scattering for high-quality acoustic rendering.

However, there is clearly still room for improvements. To retain finer structures in the BRDFs and RPFs, a different fitting model is required. Von-Mises-Fisher functions are a good fit for the filtered functions used in this paper, but we find that they are unsuitable for efficiently fitting the many thin lobes found in the complex reflection function.

Acknowlegements

The authors would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

References

[AB79] ALLEN J. B., BERKLEY D. A.: Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America 65*, 4 (1979), 943–950. 1

[AN58a] ALLRED J. C., NEWHOUSE A.: Applications of the monte carlo method to architectural acoustics. *The Journal of the Acoustical Society of America 30*, 1 (1958), 1–3. 1

[AN58b] ALLRED J. C., NEWHOUSE A.: Applications of the monte carlo method to architectural acoustics. ii. *The Journal of the Acoustical Society of America 30*, 10 (1958), 903–904. 1

[CD09] COX T. J., D'ANTONIO P.: *Acoustic absorbers and diffusers: theory, design and application*. Taylor & Francis, 2009. 1, 2

[CDD*06] COX T., DALENBÄCK B.-I., D'ANTONIO P., EMBRECHTS J.-J., JEON J., MOMMERTZ E., VORLÄNDER M.: A tutorial on scattering and diffusion coefficients for room acoustic surfaces. *Acta Acustica United With Acustica 92* (2006), 1–15. 2

[Dal96] DALENBÄCK B.-I. L.: Room acoustic prediction based on a unified treatment of diffuse and specular reflection. *The Journal of the Acoustical Society of America 100*, 2 (1996), 899–909. 1

[DES99] DINDART J.-F., EMBRECHTS J.-J., SÉMIDOR C.: Use of a kirchhoff bidirectional reflectance distribution function in a particle tracing method. In *Proceedings of the 2nd Convention Forum Acusticum* (1999). 1, 2

[DKS93] DALENBÄCK B.-I., KLEINER M., SVENSSON P.: A macroscopic view of diffuse reflection. In *Audio Engineering Society Convention 95* (10 1993). 1

[DW94] DUTRÉ P., WILLEMS Y.: Importance-driven monte carlo light tracing. *Photorealistic Rendering Techniques (Proceedings of the 5th Eurographics Workshop on Rendering)* (1994), 185–194. 6

[EAS01] EMBRECHTS J. J., ARCHAMBEAU D., STAN G.: Determination of the scattering coefficient of random rough diffusing surfaces for room acoustics applications. *Acta Acustica united with Acustica 87*, 4 (2001), 482–494. 2

[Emb00a] EMBRECHTS J.-J.: Modelling sound diffusion in ray tracing programs. In *Proceedings of Acustica 2000 - TecniAcustica Congress* (2000). 2

[Emb00b] EMBRECHTS J.-J.: Simulation of first and second order scattering by rough surfaces with a sound ray formalism. *Journal of Sound and Vibration 229*, 1 (2000), 65–87. 2

[FHLB99] FILIPPI P., HABAULT D., LEFEBVRE J.-P., BERGASSOLI A.: *Acoustics: basic physics, theory and methods*. Academic Press, 1999. 4

[HSRG07] HAN C., SUN B., RAMAMOORTHI R., GRINSPUN E.: Frequency domain normal map filtering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 3 (2007). 3

[Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH '86) 20*, 4 (1986), 143–150. 2

[KSS68] 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 (1968), 118–125. 1

[Lam96] LAM Y.: A comparison of three diffuse reflection modeling methods used in room acoustics computer models. *The Journal of the Acoustical Society of America 100*, 4, Part 1 (1996), 2181–2192. 1

[Ram09] RAMAMOORTHI R.: Precomputation-based rendering. *Foundations and Trends in Computer Graphics and Vision 3*, 4 (2009), 281–369. 2

[Sch70] SCHROEDER M. R.: Digital simulation of sound transmission in reverberant spaces. *The Journal of the Acoustical Society of America 47*, 2A (1970), 424–431. 1

[SLKS07] SILTANEN S., LOKKI T., KIMINSKI S., SAVIOJA L.: The room acoustic rendering equation. *The Journal of the Acoustical Society of America 122*, 3 (2007). 2, 6

[TDLD07] TSINGOS N., DACHSBACHER C., LEFEBVRE S., DELLEPIANE M.: Instant sound scattering. *Rendering Techniques 2007 (Proceedings of the Eurographics Symposium on Rendering)* (2007). 1, 2, 5

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 6

[WAT92] WESTIN S. H., ARVO J. R., TORRANCE K. E.: Predicting reflectance functions from complex surfaces. *Computer Graphics (Proceedings of SIGGRAPH '92) 26*, 2 (1992), 255–264. 1

[Zwi61] ZWICKER E.: Subdivision of the audible frequency range into critical bands (frequenzgruppen). *The Journal of the Acoustical Society of America 33*, 2 (1961), 248–248. 3