Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments

Nicolas Tsingos and Jean-Dominique Gascuel

iMAGIS†/GRAVIR-IMAG/INRIA

Abstract

We present a new approach to efficiently compute effects of sound occlusion and diffraction by obstacles in general virtual acoustic environments. Based on the Fresnel-Kirchoff approximation to diffraction, our method uses a 3D model of the environment and computer graphics hardware to evaluate diffraction effects between a sound source and a receiver position. No further hypothesis is needed on the diffracting domain. This method allows us to compute attenuation data and filters which can then be used for auralization. The method can also be extended to indirect sound propagation paths.

1 Introduction

Several methods have already been designed to render a virtual sound field for interactive virtual reality applications, including effects such as Doppler effect for moving sound sources and use of binaural cues or filtering techniques for convincing immersive "3D-sound". Nevertheless, existing approaches do not take properly into account effects of sound occlusion during the simulation which can lead to a significant loss of realism [9]. Computing realistic sound occlusion is a difficult problem due to the phenomenon of diffraction. In a natural environment, audible sound waves, whose wavelengths range from a few millimeters to tens of meters, are diffracted by surrounding objects. Thus, sound diffraction cannot always be neglected and computing the resulting effects usually can only be achieved using ad-hoc models or time consuming techniques.

We present a novel approach to compute sound occlusion and diffraction for fully dynamic virtual acoustic environments where sources, listeners and occluders are moving. Given a geometrical model of a virtual environment, this method allows fast calculation of sound diffraction between two points in space. We use computer graphics hardware, which becomes more and more common on various computing platforms, to reconstruct in interactive to real-time rates the diffracting domain so we can evaluate the diffracted sound field.

First, we review some previous work aimed at computing sound diffraction and recall some details about the Fresnel-Kirchoff approximation that we use in our calculation. Then, we show how the diffraction integral can be evaluated efficiently and in a generic manner using standard computer graphics hardware rendering. We present some results validating our approach in several well known cases and discuss its limitations. Finally, we show how our model can be extended to indirect sound propagation paths in an image-source calculation context and used in an auralization system.

†iMAGIS is a joint research project of CNRS/INRIA/UJF/INPG.
iMAGIS/GRAVIR, BP 53, F-38041 Grenoble cedex 09 France.
E-mail: {Nicolas.Tsingos|Jean-Dominique.Gascuel}@imag.fr
http::/www-imagis.imag.fr/

2 Previous approaches

Various methods exist to compute sound occlusion and diffraction effects, especially to study attenuation of traffic noise by acoustic screens [10]. However, these methods define the diffracted sound field only for obstacles of very simple geometry (e.g. planes, cylinders or ellipsoids). In more complex configurations, two main approaches are used: finite elements methods [15] and geometrical theory of diffraction [7, 8]. Although accurate, finite element methods suffer from their computational cost (up to hours of calculation) and thus cannot be applied to interactive or real-time applications. However, they remain the reference approaches if it comes to accuracy and generality. Geometrical theory of diffraction is an extension of geometrical optics to diffraction phenomena (through the definition of diffracted rays) and has also been applied to sound waves. But constructing the diffracted rays is a difficult task in arbitrary 3D environments and remains quite costly, if possible. Acceleration structures such as ray/beam-trees can, of course, be used in order to precompute information [6, 12] . This approach is adequate in the case of a moving listener, but part of, if not the whole structure would have to be be reconstructed if a sound source or the environment is modified [3]. Moreover, necessary diffraction coefficients must be found by a comparison to canonical cases which limits the method in the case of complex environments. Another approximation often used is the Fresnel-Kirchoff theory of diffraction [4]. Our calculation is based on this theory, presented in more detail below.

3 The Fresnel-Kirchoff diffraction theory

The Fresnel-Kirchoff approximation to diffraction is well known in geometrical optics [5, 1]. It is *a priori* also valid in the case of high-frequency sound waves. Fresnel diffraction is based on the *Huygens-Fresnel principle* stating that *every unobstructed point of a wavefront, at a given instant in time, serves as a source of spherical secondary waves (with the same frequency as that of the primary wave). The amplitude of the acoustic field at any point beyond is the superposition of all these waves (considering their amplitudes and phases)* (Figure 1). This rather hypothetical principle has been later on developed in a more rigorous way by Kirchoff, who proved that it can be derived from the scalar diffraction theory (see appendix B).

Thus, let us consider a sound source emitting a spherical, harmonic sound wave of wavelength λ and a 3D diffracting domain of envelope Σ.

The unoccluded sound pressure disturbance can be expressed as:

$$P_u(t,\rho) = \frac{\epsilon_o}{\rho} e^{i(k\rho-\omega t)} = \hat{P}_u(\rho) e^{-i\omega t},$$

where $k = 2\pi/\lambda$ is the wave number, $\omega = kc$, c is the sound propagation speed and ϵ_o is the source strength.

According to Fresnel-Kirchoff's theory, the sound pressure contribution $d\hat{P}(M)$ of a small differential area dS of the unoccluded wavefront to the total sound pressure $\hat{P}(M)$ at any point M beyond Σ can be expressed as (omitting the $e^{-i\omega t}$ term):

$$d\hat{P}(M)dS = -\frac{\epsilon_o e^{ik(\rho+r)}}{4\pi\rho r}\left(ik(1+\mathbf{n}\cdot\mathbf{r}) - \frac{1}{\rho} - \frac{\mathbf{n}\cdot\mathbf{r}}{r}\right) dS \qquad (1)$$

(see also notations in Figure 1 and appendix B).

The total pressure $\hat{P}(M)$ at point M is then:

$$\hat{P}(M) = \iint_{\Sigma_{\text{UNOCCLUDING}}} d\hat{P}(M)dS,$$

which can be also expressed as:

$$\hat{P}(M) = \hat{P}_u(M) - \iint_{\Sigma_{\text{OCCLUDING}}} d\hat{P}(M) dS, \qquad (2)$$

where $\hat{P}u(M)$ is the unoccluded sound pressure disturbance at point M.

4 Using computer graphics hardware to compute diffraction effects

Evaluating the diffracted sound field involves solving two problems:

- identify the diffracting domain between the source and the receiver, $\Sigma_{\text{OCCLUDING}}$,
- evaluate the diffraction integral (2).

In the following sections we describe how the diffracting domain can be reconstructed in real-time for an arbitrary 3D environment where sources, receivers and occluding objects are moving. We present two ways of evaluating the diffraction integral and discuss the limitations of our approach. Then, we show how our method can be applied to diffraction by screens of different geometry, including indirect propagation paths due to sound reflections, and compare our results to Boundary Element Method (BEM) simulations.

4.1 Reconstructing the diffracting domain

Given a geometrical model of a virtual environment, we can use computer graphics hardware to compute in real-time an *occlusion depth-map* of the environment between the source and the listener. To compute the depth-map, we render the scene as an image using an orthographic projection from the sound source "point of view" parallel to the source-listener direction (Figure 2). We obtain the depth of all occluders between the source and the listener along the viewing direction simply by reading the depth information stored in the Z-buffer [2], avoiding a potentially costly ray-tracing operation. Moreover, we know that "lit" pixels correspond to occluded areas (Figure 3). From the occlusion depth-map, we can reconstruct an approximation of the obstructing domain, $\Sigma_{\text{OCCLUDING}}$, by reprojecting each occluded pixel back into 3D space.

4.2 Evaluating the diffraction integral

We can therefore evaluate expression (2), the integral term being calculated as a discrete sum of differential terms (1) for every occluded pixel in the buffer leading to:

$$\hat{P}(M) = \hat{P}_u(M) - \sum_{i=1}^{Pix} \sum_{j=1}^{Pix} V(i,j) d\hat{P}_{C(i,j)}(M),$$

where $V(i,j)$ equals 1 when pixel (i,j) is lit and 0 otherwise, $d\hat{P}_{C(i,j)}(M)$ is evaluated according to expression (1) where $C(i,j)$ is the 3D point corresponding to the 2D pixel (i,j) in the depth image. The small differential area dS is approximated by the "area" of the pixel $dS = a^2/Pix^2$. a is the width of the rendering frustum and Pix is the resolution of the rendering buffer. A pseudo code of a sample calculation procedure between a source S and a receiver M is given in appendix A.

In particular cases, especially at high frequencies, when distances ρ and r are big compared to the wavelength, and when the depth of the occluding surface is nearly constant (e.g. a plane at quasi-normal incidence), expression of the diffraction integral can be simplified and an analytic solution can be derived. This solution is very common in geometrical optics and can be expressed using the Fresnel integrals [5]. In that case the depth-map can be used to find necessary integration bounds and the integration can be performed using the analytical formulas which leads to better results and makes the calculation more robust at high frequencies. However, integration bounds can only be known at the buffer resolution.

4.3 Tuning rendering parameters

One question that arises at this point is the choice of the size and resolution of the rendered occlusion depth-map. In our formulation all parts of space that do not lie in the viewing frustum are unoccluded. Thus, width of the rendered buffer must be chosen so that the entire obstacle lies in the corresponding viewing frustum. Choosing a correct resolution for the rendering buffer is non trivial. Correct evaluation of the diffraction integral will depend in a good evaluation of the length of the different occluded paths. Of course, correct treatment of interferences depends on the evaluation of this length and of the sound wavelength. Deriving a mathematical bound on the minimum possible resolution does not seem practicable, but experiments showed that $dr = \sqrt{ds} < \lambda/10$ seems appropriate to get a good treatment of interferences.

4.4 Influence of surface orientation

A problem that can be encountered, especially when using a low resolution rendering, is that of reprojection errors which cannot be neglected especially at high frequencies. Those errors are mainly due to the orientation of the occluders relative to the view direction. If the occluder's surface is tangent to the view direction, the 3D reconstruction of the corresponding pixels will be subject to errors (Figure 4). However, we can correct them using computer graphics hardware to take into account surfaces' orientations in our calculation. To achieve this we simply use the standard lambertian diffuse illumination model [2] to render our obstacles. According to this model and given a light source, the light intensity of a point on a surface is given by:

$$I = k_d(\vec{n}.\vec{l}),$$

where k_d is a shading constant, $\vec{n}$ is the normalized local normal of the surface and $\vec{l}$ is the normalized direction between the point of interest on the surface and the light source (Figure 4).

Thus, if we set $k_d = 1$ and illuminate our virtual world with a directional light source directed along view direction, the illumination value i.e. the pixel color is directly the cosine of the angle between the local normal of the corresponding 3D surface and the view direction. We can use this term to correct the reconstruction errors or as a weight to change the influence of the surfaces depending on their orientation.

Note that as this computation uses standard 3D graphics hardware capabilities, the added computation time is negligible.

4.5 Implementation

Our method has been implemented in C++ using the standard OpenGL 3D graphics library [11] and 3D graphics database tool OpenInventor [14], which makes it easily portable to a variety of platforms including PC's. The current implementation runs on Silicon Graphics workstations.

5 Results

5.1 Diffraction by simple obstacles

Figures 5 and 6 show the results of our method in two well known cases, the infinite half plane and the square aperture. Note that these examples are challenging for our method since the actual size of the occluders is ”infinite” and we only consider a finite frustum. Table 1 gives comparative execution times to compute point to point attenuation data and quantitative error in the infinite half plane case (computed on an SGI Indy R5000 150MHz workstation). Note that even at low resolution, the method gives good qualitative results.

Table 1: Comparative execution times for the half plane example at 1000Hz

resolution (pixels)	a	time (s)	error (%)
100x100	10.0	0.1	7
200x200	20.0	0.6	5
400x400	30.0	2.5	1.5
1500x1500	60.0	36	0.2

5.2 Application to visualization of sound diffraction phenomena

Since our calculation is quite fast, it is possible to compute high quality energy maps, for example, to get a visualization of the influence of the occluding object. This kind of images can be computed by simply sampling a receiving plane in a ray-tracing-like approach and performing our calculation for each sampled receiving point. Figures 7, 5 and 6 show various intensity maps in different occluded configurations. Computation time greatly depends on the resolution of the final image and the resolution of each intermediate rendering, but is usually of the order of a few hours.

5.3 Application to interactive auralization

One of the main advantages of the method is fast evaluation of diffraction without any limiting assumption on the geometry of the diffracting domain. This makes our method very appropriate for interactive graphics applications where it can be used to add occlusion and diffraction effects to the simulated sound. We compute a diffraction filter by running our calculation for different frequencies (e.g. central values of octave bands) and then use it to filter a rough sound signal. Figure 8 shows an example of attenuation spectrum calculated for an infinite half plane.

For applications where no quantitative result is needed (e.g. for interactive entertainment), a simpler method can be used. Sound attenuation that is realistic enough can be computed by evaluating the proportion of unoccluded pixels in the rendered image and using this term as an attenuation factor [13]. In order to get a frequency dependent attenuation, the width of the frustum needs to decrease as frequency increases. For very high frequencies this process will tend to a single ray-casting visibility test. One solution is to choose frustum size such that it contains the first Fresnel ellipsoid defined by the source and the receiver positions [5], i.e. $a = \sqrt{\lambda(\lambda/4 + d)}$, where d is the source to receiver distance.

6 Extension to indirect propagation paths

The proposed method has been described for direct sound propagation between a source and a receiving position in presence of non-reflective occluders. We can extend our method using an image source model to take into account indirect propagation paths (to handle reflections on a ground for example [10]). In that case, we perform a rendering for each image source and render the occluders using the appropriate mirror transformation (Figure 9).

Tables 2 to 5 show comparative results between our Z-buffer technique and a boundary element method for different frequencies (125Hz, 250Hz, 500Hz, 1000Hz) in the case of a thin screen occluder (Figure 10). Surface impedance is considered as infinite. Results shown in the tables consist of attenuation values, $Att = 20\ log(|\hat{P}|/|\hat{P}_u|)$, and module/phase of the complex pressure ratio $\hat{p} = |\hat{p}|e^{i\phi} = \hat{P}/\hat{P}_u$. Phase is expressed in degrees between 0^o and 360^o. Frustum width used for Z-buffer calculation is $a = 10\lambda$ and resolution is 1000×1000 pixels. Calculation time varies between 2 and 6 seconds for a source-receiver couple on SGI O2 workstation.

Table 2: thin screen, 125 Hz

height (m)	Att (dB)		Δ_{Att} (dB)	$(\|\hat{p}\|, \phi)$		$\Delta_{(\|\hat{p}\|,\phi)}$
	BEM	Z-buffer		BEM	Z-buffer	
0.0 (M0)	-0.10	-2.88	**2.78**	(0.987, 107.58)	(0.717, 99.57)	**(0.270, 8.01)**
0.5 (M1)	-0.14	-3.21	**3.07**	(0.984, 108.55)	(0.691, 94.14)	**(0.293, 14.41)**
1.0 (M2)	-0.24	-2.66	**2.42**	(0.973, 109.42)	(0.735, 90.52)	**(0.238, 18.90)**
1.5 (M3)	-0.39	-3.57	**3.18**	(0.955, 110.23)	(0.662, 92.11)	**(0.293, 18.12)**
2.0 (M4)	-0.62	-3.61	**2.99**	(0.931, 110.96)	(0.659, 93.12)	**(0.272, 17.84)**
2.5 (M5)	-0.91	-4.12	**3.21**	(0.900, 111.55)	(0.621, 100.43)	**(0.279, 11.12)**
3.0 (M6)	-1.28	-4.81	**3.53**	(0.863, 112.03)	(0.574, 94.48)	**(0.289, 17.55)**
3.5 (M7)	-1.72	-4.36	**2.64**	(0.820, 112.34)	(0.605, 88.98)	**(0.215, 23.36)**
4.0 (M8)	-2.24	-5.35	**3.11**	(0.772, 112.43)	(0.540, 86.76)	**(0.232, 25.67)**
4.5 (M9)	-2.85	-5.45	**2.60**	(0.720, 112.22)	(0.533, 87.66)	**(0.187, 24.56)**

Table 3: thin screen, 250 Hz

height (m)	Att (dB)		Δ_{Att} (dB)	$(\|\hat{p}\|, \phi)$		$\Delta_{(\|\hat{p}\|,\phi)}$
	BEM	Z-buffer		BEM	Z-buffer	
0.0 (M0)	-8.02	-10.65	**2.63**	(0.397, 168.90)	(0.293, 151.51)	**(0.104, 17.39)**
0.5 (M1)	-8.14	-9.06	**0.92**	(0.391, 170.74)	(0.352, 161.83)	**(0.039, 8.91)**
1.0 (M2)	-8.51	-10.38	**1.87**	(0.375, 172.34)	(0.303, 160.13)	**(0.072, 12.21)**
1.5 (M3)	-9.15	-10.81	**1.66**	(0.348, 173.63)	(0.288, 160.73)	**(0.060, 12.90)**
2.0 (M4)	-10.10	-11.12	**1.02**	(0.312, 174.44)	(0.278, 163.76)	**(0.034, 10.68)**
2.5 (M5)	-11.42	-14.26	**2.84**	(0.268, 174.43)	(0.194, 148.17)	**(0.074, 26.26)**
3.0 (M6)	-13.23	-14.77	**1.54**	(0.218, 172.89)	(0.182, 153.52)	**(0.036, 19.37)**
3.5 (M7)	-15.69	-17.47	**1.78**	(0.164, 167.91)	(0.134, 124.95)	**(0.030, 42.96)**
4.0 (M8)	-18.95	-18.35	**-0.60**	(0.113, 155.06)	(0.120, 110.92)	**(-0.007, 44.14)**
4.5 (M9)	-21.85	-20.61	**-1.24**	(0.081, 122.61)	(0.093, 74.83)	**(-0.012, 47.78)**

Table 4: thin screen, 500 Hz

height (m)	Att (dB)		Δ_{Att} (dB)	$(\|\hat{p}\|, \phi)$		$\Delta_{(\|\hat{p}\|,\phi)}$
	BEM	Z-buffer		BEM	Z-buffer	
0.0 (M0)	-10.30	-11.59	**1.29**	(0.305, 172.25)	(0.263, 175.53)	**(0.042, -3.28)**
0.5 (M1)	-10.72	-13.62	**2.90**	(0.291, 175.75)	(0.208, 172.76)	**(0.083, 2.99)**
1.0 (M2)	-12.05	-13.77	**1.72**	(0.250, 178.17)	(0.205, 172.27)	**(0.045, 5.90)**
1.5 (M3)	-14.63	-17.04	**2.41**	(0.186, 178.13)	(0.141, 169.47)	**(0.045, 8.66)**
2.0 (M4)	-19.30	-22.95	**3.65**	(0.108, 169.57)	(0.071, 146.14)	**(0.037, 23.43)**
2.5 (M5)	-25.22	-22.58	**-2.64**	(0.055, 114.21)	(0.074, 72.62)	**(-0.019, 41.59)**
3.0 (M6)	-19.12	-19.34	**0.22**	(0.111, 59.37)	(0.108, 50.73)	**(0.003, 8.64)**
3.5 (M7)	-14.31	-16.69	**2.38**	(0.192, 51.07)	(0.146, 47.55)	**(0.046, 3.52)**
4.0 (M8)	-11.57	-13.54	**1.97**	(0.264, 51.28)	(0.210, 48.98)	**(0.054, 2.30)**
4.5 (M9)	-10.02	-10.74	**0.72**	(0.315, 53.60)	(0.290, 55.47)	**(0.025, -1.87)**

Table 5: thin screen, 1000 Hz						
height (m)	Att (dB)		Δ_{Att} (dB)	$(\lvert\hat{p}\rvert, \phi)$		$\Delta_{(\lvert\hat{p}\rvert,\phi)}$
	BEM	Z-buffer		BEM	Z-buffer	
0.0 (M0)	-12.30	-12.95	**0.65**	(0.243, 78.91)	(0.225, 61.57)	**(0.018, 17.34)**
0.5 (M1)	-14.14	-14.63	**0.49**	(0.196, 85.86)	(0.186, 67.61)	**(0.010, 18.25)**
1.0 (M2)	-22.40	-24.26	**1.86**	(0.079, 85.45)	(0.062, 77.78)	**(0.017, 7.67)**
1.5 (M3)	-22.03	-22.55	**0.52**	(0.079, 302.33)	(0.075, 302.08)	**(0.004, 0.25)**
2.0 (M4)	-14.02	-14.67	**0.65**	(0.199, 302.67)	(0.185, 303.88)	**(0.014, -1.21)**
2.5 (M5)	-12.21	-12.19	**-0.02**	(0.245, 309.49)	(0.245, 303.34)	**(0.000, 6.15)**
3.0 (M6)	-13.95	-15.34	**1.39**	(0.201, 314.58)	(0.171, 304.24)	**(0.030, 10.34)**
3.5 (M7)	-20.77	-22.43	**1.66**	(0.091, 301.66)	(0.076, 282.11)	**(0.015, 19.55)**
4.0 (M8)	-20.02	-20.40	**0.38**	(0.099, 207.98)	(0.096, 204.59)	**(0.003, 3.39)**
4.5 (M9)	-13.41	-14.42	**1.01**	(0.213, 198.06)	(0.190, 194.97)	**(0.023, 3.09)**

7 Discussion

Our method gives good qualitative results in environments of arbitrary geometry. Moreover, comparisons to BEM simulations in several configurations with obstacles of infinite extent, presented above for a thin infinite screen, give satisfactory quantitative results. The obtained values differ because our method considers the obstacles only within the finite frustum. Taking a larger frustum and increasing the rendering resolution leads to more precise simulations but slows down the computation process. Also, our approach seems to be more suitable for high frequencies.

Only the obstacles "visually" visible from the source point of view are taken into account. This means that when several obstacles are overlapping, only the one closest to the source will be taken into account.

Finally our method does not allow to simulate surfaces of finite impedance.

8 Conclusion and future works

We have presented an approach to efficiently approximate sound occlusion effects based on the Fresnel-Kirchoff theory of diffraction. Our method is fast and can be applied to general environments thanks to the use of graphics hardware rendering. This makes it well suited to interactive auralization or visualization of diffracted energy maps. It can also be extended to indirect sound propagation paths. Further investigations shall include extension of the method to surfaces of finite impedance and overlapping obstacles. More BEM comparisons also need to be performed in order to validate our method for more complex occluders.

Acknowledgments

The authors would like to thank the "Acoustic and electromagnetic waves" team at the Centre Scientifique et Technique du Bâtiment in Grenoble, in particular Yannick Gabillet and Jerôme Defrance, for helpful chats and help with the validation process against BEM simulations.

References

[1] D. Dauger. Simulation and study of fresnel diffraction for arbitrary two dimensional apertures. *Computer in physics*, 1996.

[2] Foley, VanDam, Feiner, and Hughes. *Computer graphics, principles and practice*. Addison Wesley, 1990.

[3] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West. A beam tracing approach to acoustic modeling for interactive virtual environments. *http:://www.cs.princeton.edu/˜funk/rsas.html*, 1998.

[4] Y. Furue. Sound propagation from the inside to the outside of a room through an apperture. *Applied Acoustics*, 31(1), 1990.

[5] E. Hecht. *Optics, second edition*. Addison Wesley, 1987.

[6] P. Heckbert and P. Hanrahan. Beam tracing polygonal objects. *ACM Computer Graphics, SIGGRAPH'84 Proceedings*, 18(3), 1984.

[7] T. Kawai. Sound diffraction by a many sided barrier or pillar. *Journal of Sound and Vibration*, 79(2), 1981.

[8] J.B. Keller. Geometrical theory of diffraction. *Journal of the Optical Society of America*, 52(2), 1962.

[9] M. Kleiner, B. Dalenbak, and P. Svensson. Auralization - an overview. *Journal of the Audio Engineering Society*, 41(11), November 1993.

[10] Z. Maekawa. Noise reduction by screens. *Applied Acoustics*, 1, 1968.

[11] J. Neider, T. Davis, and W. Mason. *OpenGL Programming Guide*. Addison-Wesley, 1993.

[12] S.J. Teller. *Visibility Computations in Densely Occluded Polyhedral Environments*. PhD thesis, Computer Science Division (EECS), University of California, Berkley, 1992.

[13] N. Tsingos and J.D. Gascuel. Soundtracks for computer animation : sound rendering in dynamic environments with occlusions. *Proceedings of Graphics Interface'97*, 1997.

[14] J. Wernecke. *The Inventor Mentor*. Addison-Wesley, 1994.

[15] J.R. Wright. Fundamentals of diffraction. *Journal of the Audio Engineering Society*, 45(5), may 1997.

A Pseudo-code for Fresnel diffraction calculation

```
complex FresnelDiffraction( point S, M ; real λ ; int a, Pix)

{
  setupCamera(S,M,Pix,a)
  render()
  real d = norm(S-M)
  real k = 2π/λ
  complex Punoccluded = e^{ikd}/d
  for int i = 0 to Pix
    for int j = 0 to Pix
      if isPixelOccluded(i,j)
      {
        point C = unprojectPixel(i,j)
        point R = M - C
        point N = C - S
        real r = norm(R)
        real ρ = norm(N)
        normalize(N)
        normalize(R)
        complex deltaOccluded = dP̂(k,r,ρ,R,N)a^2/Pix^2
        Punoccluded = Punoccluded - deltaOccluded
      }
  return Punoccluded
}
```

B Helmoltz equation and the Kirchoff integral theorem

In this section we quickly recall how the differential pressure disturbance term used in our calculation can be derived from the scalar field diffraction theory. Consider a sound source and a receiving location. The sound pressure disturbance at the receiver is a solution of the wave equation:

$$\nabla P = \frac{1}{c^2}\frac{\partial^2 P}{\partial t^2}$$

We can express the solution on the form $P = \hat{P}e^{-ikct}$ (where c is the sound speed). Substituting this in the wave equation we obtain the *Helmoltz equation*:

$$\nabla^2 \hat{P} + k^2 \hat{P} = 0.$$

Solving this equation with the help of Green's theorem leads to an expression of the pressure disturbance at point M in terms of the pressure disturbance and its gradient evaluated on an arbitrary closed surface S, enclosing M (Figure 11):

$$\begin{aligned} \hat{P}(M) = \quad & \frac{1}{4\pi}\left[\iint_S \frac{e^{ikr}}{r}\nabla \hat{P}\cdot d\mathbf{S} \right. \\ & \left. - \iint_S \hat{P}\nabla\left(\frac{e^{ikr}}{r}\right)\cdot d\mathbf{S}\right], \end{aligned} \tag{3}$$

which is known as the *Kirchoff integral theorem*.

If we apply this theorem to an unoccluded spherical wavefront, sound pressure has the form $\hat{P}(\rho) = \frac{\epsilon_o}{\rho}e^{ik\rho}$. Substituting into equation (3) and choosing a proper integration surface leads to (Figure 1):

$$\begin{aligned} \hat{P}(M) = \quad & -\frac{\epsilon_o}{4\pi}\left[\iint_S \frac{e^{ik(\rho+r)}}{\rho r} \right. \\ & \left. \left(ik(1+\mathbf{n}\cdot\mathbf{r}) - \frac{1}{\rho} - \frac{\mathbf{n}\cdot\mathbf{r}}{r}\right) dS\right], \end{aligned} \tag{4}$$

where S is the surface of the wavefront itself (see [5] for a more detailed explanation.)

We can rewrite equation (4) as:

$$\hat{P}(M) = \iint_S d\hat{P}(M)dS$$

leading to a differential pressure disturbance $d\hat{P}(M)$ equal to:

$$d\hat{P}(M) = -\frac{\epsilon_o e^{ik(\rho+r)}}{4\pi\rho r}\left(ik(1+\mathbf{n}\cdot\mathbf{r}) - \frac{1}{\rho} - \frac{\mathbf{n}\cdot\mathbf{r}}{r}\right)$$

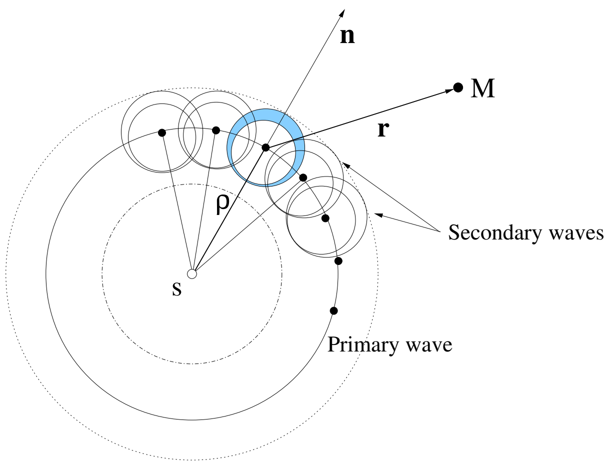


Figure 1: *The Huygens-Fresnel principle. A sound wave can be represented as a sum of secondary waves of same frequency considering their amplitudes and phases.*

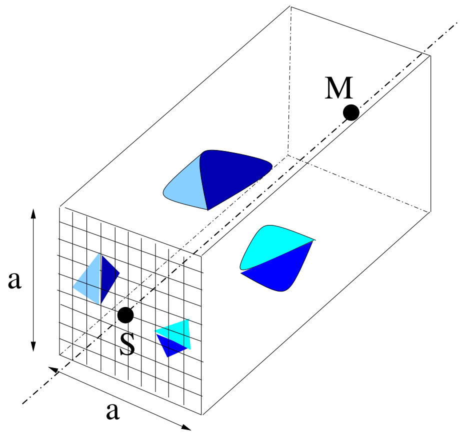


Figure 2: *Viewing frustum for occlusion depth-map calculation. A parallel projection is used to compute a depth picture of the occluding objects between sound emitter and receiver positions.*

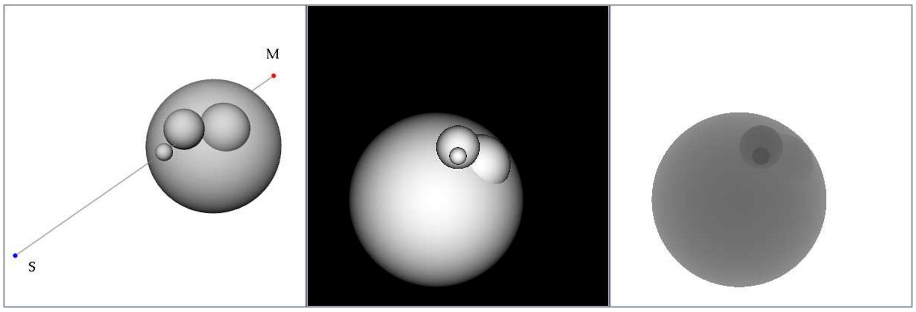


Figure 3: *Example of rendered buffers. Left picture shows a 3D scene with a source (S) in the foreground, spherical occluders and a receiver (M) in the background. Center picture correspond to the orientation map from the source point of view. Darker pixels corresponding to surfaces tangent to the view direction. Right picture shows the corresponding occlusion depth-map buffer. Darkest grey shades correspond to points closest to the source along source-to-receiver direction.*

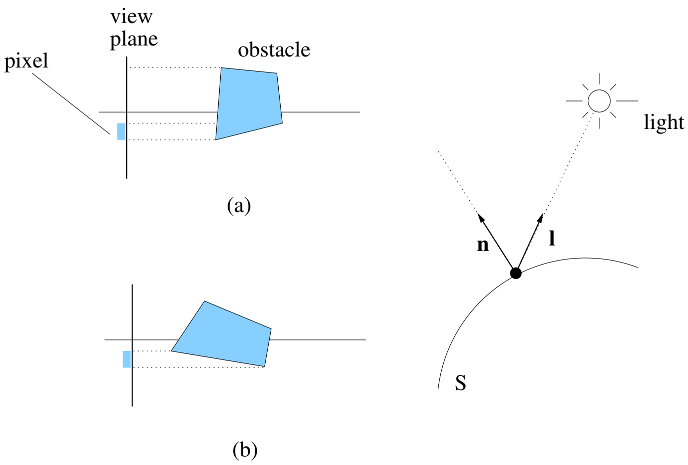
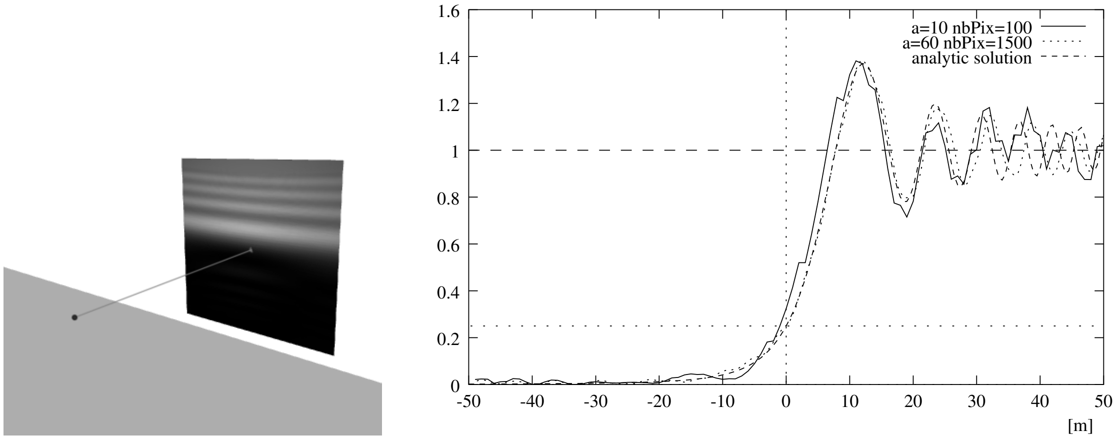


Figure 4: *Taking into account occluders orientation. (a) and (b) shows two different configurations for an occluded pixel. In the (b) case, reprojection will be subject to errors because a larger 3D surface is projected onto a single pixel. Right-hand side figure shows notations for lambertian diffuse illumination model.*



(a) (b)

Figure 5: *(a) Diffraction pattern (intensity map) for an infinite half plane at 1000 Hz. Sound source to diffracting plane distance is 20 m. Diffracting plane to receiving plane distance is 100 m.Receiving screen is* $100 \times 100m^2$ *wide. (b) Variations of the intensity ratio between occluded and unoccluded sound depending on the vertical position (in meters. 0 corresponds to a receiver location aligned with the sound source and edge of the diffracting plane) in the receiving plane for different values of the frustum width* a *and rendering resolutions.*

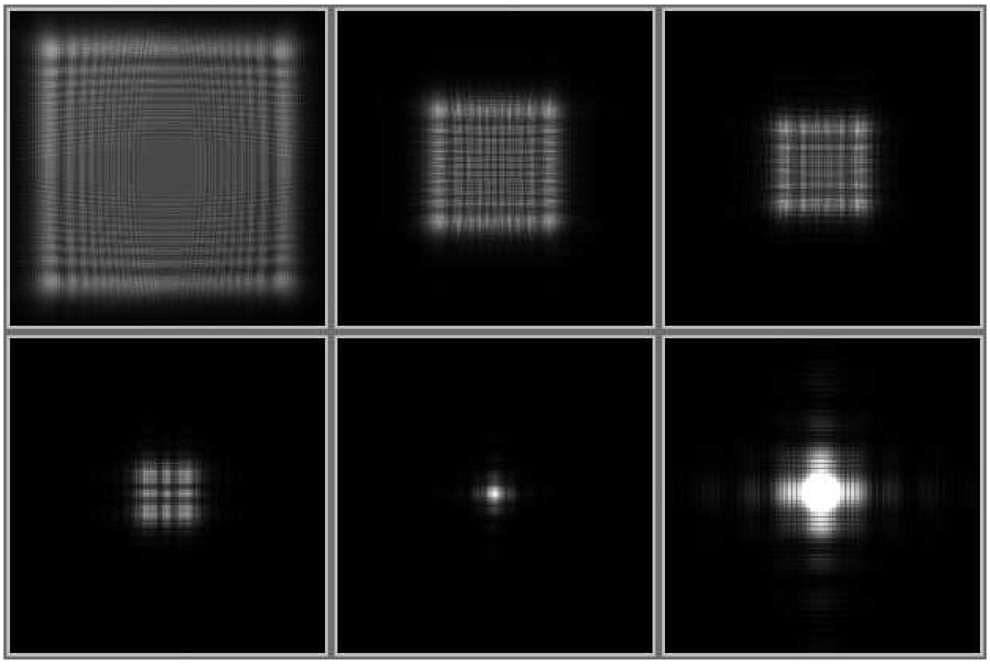
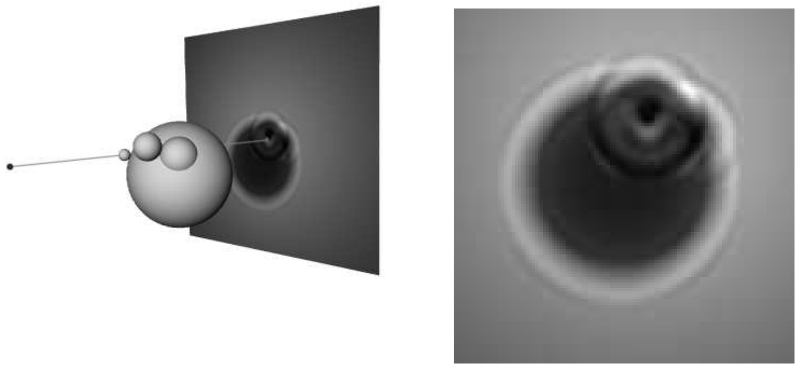


Figure 6: *Diffraction patterns (intensity maps) for a plane square aperture of decreasing size. Same experimental conditions as Figure 5. Aperture size ranges from* $15\times15m^2$ *(top-left) to* $2\times2m^2$ *(bottom-right). Notice the change between near field (Fresnel) and far field (Fraunhofer) diffraction patterns as aperture size is decreasing. Last image is a close up of the Fraunhofer-like diffraction pattern.*



(a) (b)

Figure 7: *(a) Example of irradiance pattern on a receiving square screen beyond the occluding domain computed with a* 100×100 *pixels buffer and* $a = 4.0$. *Source to screen distance is 120m. Receiving screen is 100m wide. Frequency is 1000Hz. (b) Close-up of the diffraction pattern. The pattern is computed at* 200×200 *pixels resolution, applying our point-to-point method for each pixel. Computation time is 2 hours (on SGI O2 180 MHz R5000 workstation) which gives an average time of 0.02s per pixel.*

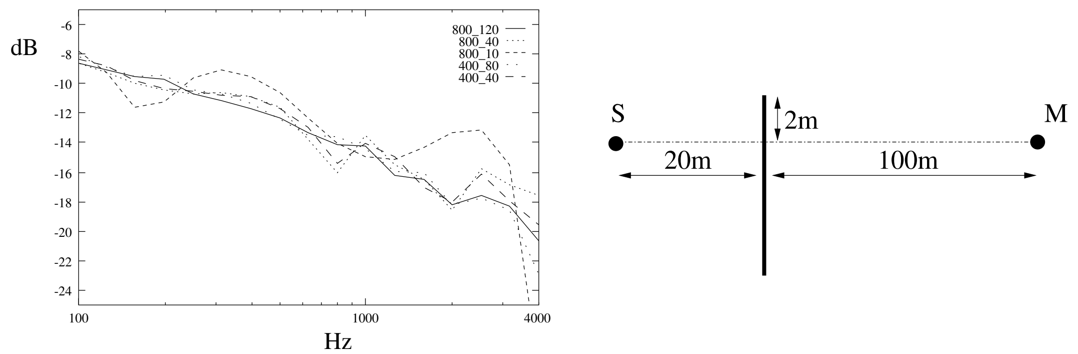


Figure 8: *Attenuation spectra for an infinite half plane for different values of* a *and buffer resolution* res *(labeled* res_a*). Relative positions of source, receiver and screen are given on the right-hand side figure*

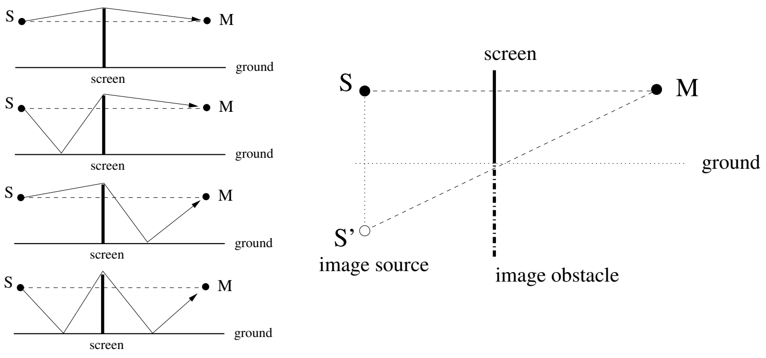


Figure 9: *Extension to indirect propagation paths. To take into account the effect of sound reflexion on the reflective ground, another rendering is performed from an image source point of view. A complementary "image-obstacle" is also added to the scene. Left-hand side figure shows the equivalent simulated propagation paths.*

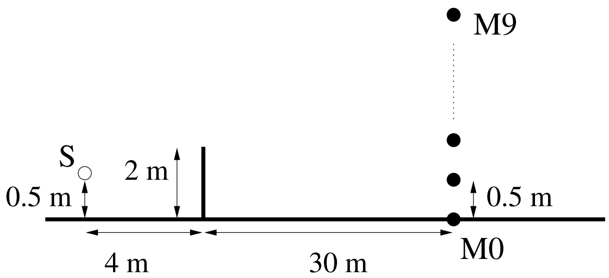


Figure 10: *Configuration used for BEM comparisons. Figure show the 2D profile of the terrain. Screen is infinitely wide.*

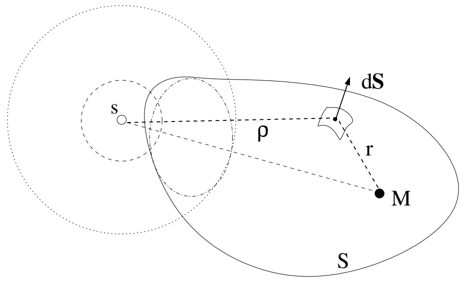


Figure 11: *The Kirchoff integral theorem. Solving the Helmoltz equation using Green's theorem.*