Real-Time Implementation of a General Model for Spatial Processing of Sounds

Shahrokh Yadegari
sdy@ucsd.edu

F. Richard Moore
frm@ucsd.edu

Harry Castle
hcastle@ucsd.edu

Anthony Burr
aburr@ucsd.edu

Ted Apel
tapel@ucsd.edu

Abstract

In 1982, one of the authors proposed a general model for spatial processing of sounds and included a partial implementation of it in the `cmusic` sound synthesis program[7]. This model is designed to simulate the most perceptible physical characteristics of a real or imaginary space relating to the localization of sound sources. The original model defines the acoustic space in two dimensions as an outer closed space (the illusory acoustic space) and an inner listening room (the intended performance space) with "openings" along its perimeter in the location of the speakers. The spatial impression is produced by simulating the direct radiation, early echoes, and global reverberation of a sound source as heard through each opening. This paper discusses our modifications of the original `cmusic` implementation: first, the algorithm runs in real-time, and second, we made additions that more satisfactorily realize the general model. Our implementations of the algorithm in Pd and Max/MSP are presented.

1 Introduction

This paper describes a real-time implementation of the "general model for spatial processing of sounds" proposed by one of the authors[7], who made a partial, though fairly complete, implementation of it as part of the `cmusic` package[6]. The general model draws on basic psychophysics of spatial perception (mostly the work of Gardner[4], Blauert[2], and Stevens[11]) and on work in room acoustics, particularly the work of Schroeder[9] and Moorer[8] in developing realistic artificial reverberation.

The central conceit of the algorithm is that of a "room within a room". The inner room is the space delimited by the speakers which contains the listeners. The model simulates the behavior of the sound source within a user-defined virtual outer room, as heard from the inner room. The speakers act as the "windows" through which sound from the outer room passes. The greatest strength of the algorithm, however, is that its spatial effects are minimally position dependent across an audience. This is achieved by focusing on simulating the physical characteristics of the real or imaginary space as heard in a defined performance space rather than from vantage point of a single listener.

2 Moore's Spatial Algorithm

Once the room dimensions are defined, the algorithm scales amplitude of the source signal to account for the directionality of the source. It then applies basic ray-tracing principles to simulate direct paths from the source to each speaker and early reflections of the source from each surface of the outer room to each speaker.(See Figure 2.)

2.1 Sound Source Model

The model uses three variables to simulate the radiation of the sound from each point source. The implementation of Moore's spatial model in `cmusic` (called the "space unit generator") receives a single audio input, and it allows for an arbitrary numbers of radiation vectors (RV) as follows:

$$RV = (x, y, \theta, amp, back), \qquad (1)$$

where x and y are the location coordinates in meters with (0,0) at the center of the inner room, θ is the direction of the vector, amp is the length of the vector, and $back$ is the relative radiation factor in the opposite direction of $\theta (0 \leq back \leq 1)$. The following equation generates the amplitude scale factors

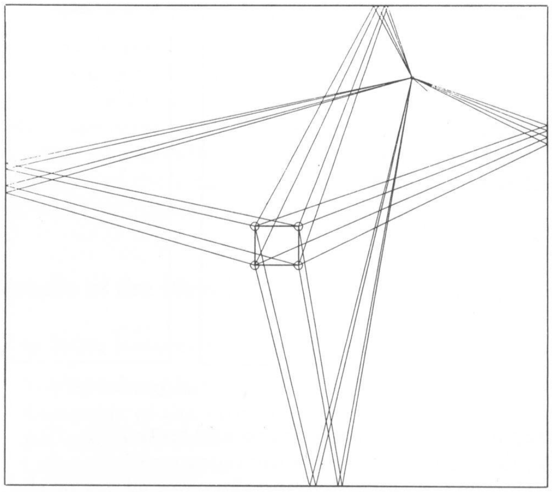

Figure 1: First order reflections off the four walls to 4 speakers.

which simulate a hypercardioid pattern:

$$r(\phi) = \left[1 + \frac{(back - 1)\,|\theta - \phi|}{\pi}\right]^2 \qquad (2)$$

where $r(\phi)$ is the scalar for the simulated ray. θ is the source radiation direction, ϕ is the direction of the ray being simulated, and $back$ is the relative radiation factor in the opposite direction of θ ($0 \leq back \leq 1$).

2.2 Direct and Reflected Path Calculations

Direct paths are calculated simply as straight lines from the source to each of the speakers and early reflections are calculated as two-part lines from the source reflected by the walls to each speaker. The differing impressions of the source at each speaker are accounted for by separate taps into a single delay line.

For direct paths, the distance from source to each speaker determines the delay time of that channel and influences the amplitude in accordance with the inverse power law connecting intensity with distance

Some psychoacoustic research suggests that an inverse cubic proportion may provide a more realistic impression[10].

Early reflections provide one of the most important cues for source localization in space. The attenuation and delay times of the reflected rays are calculated according to the same algorithm used for direct paths. Moore implemented only the simulation of first order reflections in `cmusic` . The general model accounts for frequency–dependent attenuation of both direct and reflected rays due to absorption by air and the outer walls. In the `cmusic` implementation, Moore omitted any frequency-dependent attenuation, since he found that the loss of high frequencies made the spatial impression less distinct[5, p99].

The model considers the outer walls of the inner room to be completely absorptive, which aids in clarity of spatial impression. This means that an essential condition for calculating both direct paths and early reflections is whether they "cut" the inner room. If the algorithm determines that a path "cuts" the inner room, that ray is discarded.

Chowning points out that the pitch change due to Doppler effect is one of the most significant cues for sound movement in space[3]. Doppler pitch changes are a natural result of the delay set up in Moore's model.

3 New Modifications

In 1994, Ballan, Mossoni, and Rocchesso simplified Moore's spatial model for a moving sound source to run in real-time. Some of their simplifications include: (1) omitting direct ray calculation to the speaker located in the opposite quadrant of the source, (2) omitting rays reflected from any wall to the opposite speakers, and (3) calculating attenuation factors based on a simplified algorithm ($amplitude = 1 - distance$). They observed: "These are strong simplifications, but the results still give a good spatial impression."[1, p 476] We have achieved real-time performance without compromising the original model, and have added algorithms which implement the model more realistically.

3.1 Modifications for Real-time Implementation

In our implementation we have not changed the original model; however, we have optimized and simplified the calculation schemes. A computationally expensive aspect of Moore's spatial processing model is the calculation of delay times for direct and reflected sound rays. Once these values are calculated, there is no need to re-calculate them if the source does not move from its previous position. Moore's model defines an inner room which does not allow sound to travel through its walls (although the source can). The computation of whether a direct or reflected sound ray intersects any one of these walls is another computationally expensive calculation. We have achieved real-time performance by modifying the original `cmusic` implementation in two ways. First, by down-sampling the path of sound sources, and interpolating the delay times, and second, by improving inner room ray intersection detection algorithm. In this section we present and discuss the implementation of these two modifications.

3.1.1 Interpolation of Delay Times

In order to avoid discontinuities in the output signals, control signals to the space unit generator need to be continuous signals at audio rate. As mentioned above, the most computationally expensive part of Moore's algorithm is the calculation of delay times. In order to achieve real-time performance we down-sample the x and y signals to control rate kr (which means we are considering only the last coordinate in every block), and interpolate the delay times and attenuation factors within each block.

3.1.2 Errors Introduced by the Interpolation Process

The error in the interpolation process introduces non-linear and non-realistic effects in the generated output, the most prevalent of them being in estimating the position of the source and in introducing subtle timing asynchronicities among the direct and reflected rays. When our block size is 64 samples, which is less than 1.5 milliseconds at $R = 44,100\ Hz$, if a sound source is moving at 10 m/s, our error in localization will be less than 1.5 cm ($1.5\ ms * 10\ m/s = 1.5\ cm$). One way to interpret this error is that as the sound source moves faster, our interpolation algorithm would change a sound source from a point source to a small spherical source.

No matter what type of interpolation we use, we will not be able to simulate the synchronization of all the direct and reflected rays. For example, if a sound source traverses a straight line during a single block, whose two interpolating poles have the same distance to one of the speakers, the interpolation process would keep the delay time and attenuation factor for the direct ray constant for that block. Had we not used an interpolating process, the delay time would increase or decrease and return to the original value. In such a situation, the delay time and attenuation factors for the direct ray to other speakers as well as first order reflections to all speakers are likely to be interpolated toward new values.

Empirically we know that errors discussed in this section are very small and in most cases inaudible. The strength of Moore's spatial model is in localizing sound sources outside of the inner room where sound sources are considerably farther away from the listener, and therefore, more tolerant of localization and ray tracing errors.

3.2 Improved Ray Intersection Algorithm

As mentioned above, the original Moore's spatial model defines an inner room which does not allow sound to travel through its walls. The `cmusic` implementation did not account for any diffraction of sound and used a fixed 50 millisecond crossfade time for turing sound rays on and off. Moore noted that a more sophisticated approach would be to model the diffraction of the path around the edge of the cutting surfaces[7, page 13].

Diffraction of sound is a frequency dependent phenomenon; however, we implemented a simple diffraction model which produces attenuation values based on where a sound ray intersects a wall. Thus, the crossfade periods become related to the speed of the sound source. We used the distance from where a sound ray intersects a wall to the corner of the wall as a measure to calculate the diffraction attenuation factor according to the following equation:.

$$da = \begin{cases} 0 & \text{if } (TH < ds) \\ \left(\frac{TH - ds}{TH}\right)^{C} & \text{if } (0 < ds < TH) \end{cases} \tag{3}$$

where *da* is the diffraction attenuation factor, *ds* is the distance from where the the sound ray intersects a wall to the corner of the wall, *TH* is the diffraction threshold constant, and *C* is the curve of crossfade values.

In subjective listening to the results, we found the spatialization effects to sound more realistic than the results obtained by the original `cmusic`'s ray intersection detection implementation. We also found the two variables *TH* (diffraction threshold) and *C* (diffraction curve), in conjuction with *Direct* and *Reflect* variables, useful for fine tuning the spatialization effects in different performance spaces[1]. We

[1] *Direct* and *Reflect* are original user defineable `cmusic` variables which define the power laws for calculating distant attenuation values of direct and reflected rays.

have also improved the performance of the ray intersection algorithm by rewriting it with more simple arithmetic operations compared to the `cmusic` version.

3.3 Additional Improvements

We have also improved the quality of the interpolation of audio delay line for simulating the direct and reflected sound rays. The original `cmusic` implementation used linear interpolation. At fairly high speeds of the sound source one can hear certain artificial artifacts (such as high frequency noise) introduced by the spatialization process. We found that by using a 4 point interpolation we could obtain results with less distortion.

4 Summary and Conclusion

We have seen that by making changes to the `cmusic` space unit generator, we are able to run Moore's spatial algorithm in real-time. Our major modification was to interpolate the delay times within calculation blocks. We have also modified the ray intersection detection algorithms to better match the original model and have improved its performance. Successful implementations of the algorithm have been ported to the Pd and Max/MSP environments. Our implementation currently runs in a quadrophonic configuarition on Pd, under Linux RedHat 7.2 on an Intel Pentium III 866 MHz, utilizing 53% of the CPU (including the overhead of Pd), and on Max/MSP under MaxOS 9.1 on a PowerMac G4 400 MHZ utilizing 37% of the CPU (including the overhead of MAX/MSP).

5 Acknowledgments

We would like to thank Miller Puckette for his help and participation in this project.

References

[1] Oscar Ballan, Luca Mozzoni, and Davide Rocchesso. Sound spatialization in real time by first-reflection simulation. In *International Computer Music Conference*, pages 475–476, Aarhus, Denmark, 1994. ICMA.

[2] Jens Blauert. *Spatial hearing : the psychophysics of human sound localization*. MIT Press, Cambridge, Mass., rev. edition, 1997. Jens Blauert. Includes bibliographical references (p. [423]-480) and indexes.

[3] J. M. Chowning. The simulation of moving sound sources. pages 692–4, 1970. Audio Engineering Society 39th Convention New York, NY, USA 13-15 Oct. 1970 J. Audio Eng. Soc. (USA).

[4] M. B. Gardner. Binaural detection of single frequency signals in presence of noise. *Journal of Acoustical Society of America*, 34:1824–1830, 1962.

[5] F. R. Moore, M. V. Mathews, and J. R. Pierce. Spatialization of sounds over loudspeakers. In *Current directions in computer music research*, pages 89–103. MIT Press, Cambridge, MA, USA, 1989. vi+432.

[6] F. Richard Moore. The computer audio research laboratory at UCSD. *Computer Music Journal*, 6(1):18–29, 1982.

[7] F. Richard Moore. A general model for spatial processing of sounds. *Computer Music Journal*, 7(6):6–15, 1983.

[8] James A Moorer. About this reverberation business. *Computer Music Journal*, 3(2):13–28, 1979.

[9] M. R. Schroeder. Binaural dissimilarity and optimum ceilings for concert halls: more lateral diffusion. *Journal of Acoustical Society of America*, 65:958, 1979.

[10] S. S. Stevens. Neural events and the psychophysical law. *Journal of Acoustical Society of America*, 27:815, 1955.

[11] S. S. Stevens. Neural events and the psychophysical law. *Science*, 170:1043, 1970.