# Virtualized Audio as a Distributed Interactive Application

Peter A. Dinda
Department of Computer Science
Northwestern University
1890 Maple Avenue
Evanston, IL 60201
847-467-7859
pdinda@cs.nwu.edu

January 24, 2001

## Abstract

The goal of virtualized audio is to extract sound sources (lecturers, meeting participants, singers, musical instruments) from their native acoustical spaces, and insert them into a virtual acoustical space that is shared by a number of listeners. One example of virtualized audio would be to extract a string quartet from a concert hall and introduce it into one's living room. Another example would to permit geographically separated meeting participants to appear to be seated around one's own conference table. This paper describes how this problem can be broken down into two subproblems (separation and auralization), both of which can be approached in physically realistic ways if we suppose that we can draw on the distributed computational and communications resources of computational grids. Unlike traditional grid applications, however, virtualized audio requires interactive responsiveness.

## 1 Introduction

Audio systems are largely stuck in the stone ages. Their history is so long, the typical user's experience of them is so second-nature, and the field's performance metrics are so deeply ingrained that we have become largely deaf to their shortcomings. However, as we attempt to add audio to new modes of interaction such as Access Grids, their limitations are becoming more clear. Why can we not make it sound as if a remote member is sitting at the same conference table as the rest of the team? Why can should it not be possible to transport a musical performer to the listener's room? These are the same question, only the audiences are different.

An audio system is a long and complex chain of processing steps (in electronic, acoustic, and mechanical guises) that lead from a sound-producing vibration to the hairs of the listener's inner ear. In traditional audio systems, these steps largely destroy information about the original sources or confuse such information about them with other extraneous information. By the time an audio signal reaches the listener's ear, it has been changed by two different rooms, a (probably overzealous) sound engineer's mixing board, the distortion characteristics of a loudspeaker, and a feeble attempt to reconstruct one point of the original sound field in a completely different room! Is it any surprise that we feel like we are living inside the musician's guitar, or that our remote team member's voice appears to come, poltergeist-like, from no particular place?

It does not have to be this way. This paper outlines *virtualized audio*, an approach to audio that could let us transport the performer or the team member to our venue. More generally, the goal of virtualized audio is to permit listeners and performers to inject themselves into a shared virtual acoustic space—to let a listener hear what a performer would sound like in his room.

To achieve the vision of virtualized audio requires solving two problems: source separation, which we refer to as the reverse problem, and auralization, which we call the forward problem. It is important to note that neither source separation nor auralization is a new problem. My proposal is that we apply the full computational force of distributed computing environments such as the Grid to solve them better and to make them practical for users with a minimum of additional equipment. Ideally, virtualized audio would require only that a user have a collection of high quality microphones and a high quality set of headphones. However, the virtualized audio model is not limited to this equipment, and we will consider several extensions in this paper.

The main technical innovation discussed here is that of doing auralization using a physical simulation of the acoustic wave equation. This provides a clear example of how the virtualized audio approach can leverage the computational resources of the Grid. It is also an example of an interactive application that the Grid will likely have to

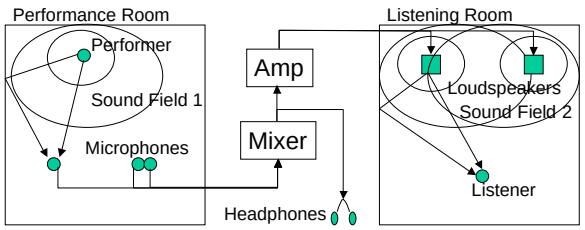


Figure 1: Traditional audio: model and acoustics.

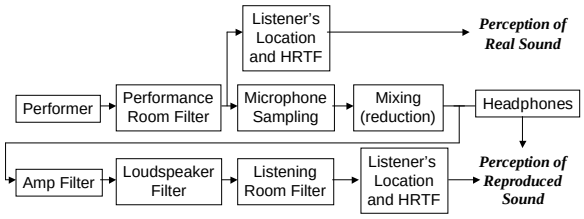


Figure 2: Traditional audio: filtering and mixing.

support in the future. Learning how to map such applications to shared distributed computing environments is a primary research focus of the author.

*Due to the extremely limited preparation time for this paper, it does not contain citations to prior work and art, and it talks quite generally about some rather complex topics. Where possible, I have tried to at least outline what is known and what can be known. In its present form, the paper is intended to be a discussion starter, not an all encompassing survey or research paper.*

## 2 Traditional audio

Figure 1 shows the structure of a traditional audio system. In this and other figures, we shall assume that digital/analog conversion is implicit and does not affect the signal. Further, we shall not consider the storage of audio streams in either analog or digital forms.

There are several important things to note about the figure. First, there is no relationship between the sound sources in the performance room (where the musician or team member is) and the sound sources in the listening room. Second, the rooms are different. The microphones pick up the sound field of the performance room, which conflates both the sounds the performer makes and the "sound" of the room. The listener hears this combination, further adulterated by the "sound" of the listening room (and his speakers—speakers are notoriously "colored" devices.) Finally, the mics in the performance room and the speakers in the listening room are intermediated by a mixer. The mixer's operator combines the audio streams picked up by the mics (48 or more for a typical in-studio music recording) into one or two streams. Notice that even if we listen via headphones, what we hear is still subject to everything up to the amplifier.

Figure 2 provides another way of looking at the traditional audio system, namely, a filter chain operating on the acoustic signal emitted by the performer. We'll assume that the filters are LTI. Although this is not true in general, it is a useful approximation. En route from the performer to a particular microphone, the signal flows through the performance room filter, which imparts it with the echos and absorption behavior of the room. If the listener were at the same position as the mic, he would hear this signal filtered through his HRTF. An HRTF, or Head Related Transfer Function, describes how an individual's upper torso, head, and outer ear modify the sound. HRTFs are critical for the perception of proper sound. The lack of an HRTF is one of the reasons why a typical recording, when heard through headphones, sounds odd.

In the case of the microphone, the signal is filtered by the characteristic of the mic. This is usually a noop, as microphones are surprisingly accurate devices, except when driven to overload. The now electronic signal is next filtered and merged with other signals from other microphones by the mixer. The output of this stage is effectively what is experienced by a headphone listener. For the speaker listener, the mixed signal is next filtered by the amplifier being used to drive his loudspeakers, and then also by the speakers themselves. It is also common to view an amplifier and a speaker as a single filter, as they are not independent. The loudspeaker launches an acoustic signal, which is filtered by the listening room. Finally, it is filtered by the listener's HRTF and turned into perception.

The important point to take away from this discussion is that the signal produced by the performer is molested early and often on its way to the listener's ears. Most of the filters through which it flows are artificial and have been placed there for convenience or because of historical limitations. The most important of these is the mixer. In the mid 1950s, it was just possible to place two channels of audio on an LP record with 25 dB of separation, and a mixer was a necessity. In the early 2000s, we can support an essentially arbitrary number of channels. Mixing down to two channels, or six, is an information crime—it destroys the richness of the performer in an effort to make it easier to introduce a threadbare facsimile of him in the listening room.

A binaural audio system, which entails micing the performance room with a dummy head sporting strategically placed microphones, and then listening to the result, without any mixing, via in-ear-canal headphones, cleanly avoids manipulation of the signal and allows the listener to insert himself into the performance room in place of the dummy head. Although this technique is at least 50 years old, it is perhaps not surprising that it has never become

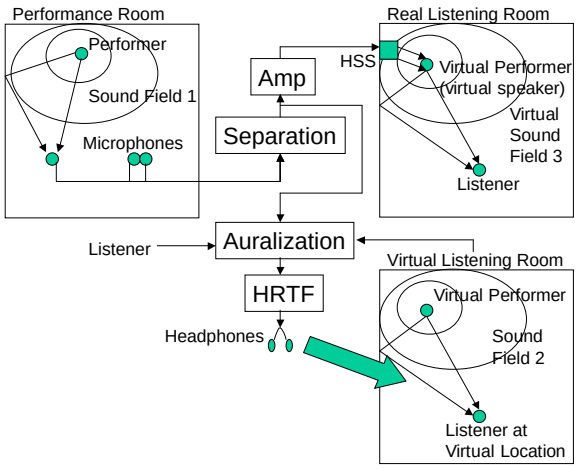


Figure 3: Virtualized audio: acoustics and model.

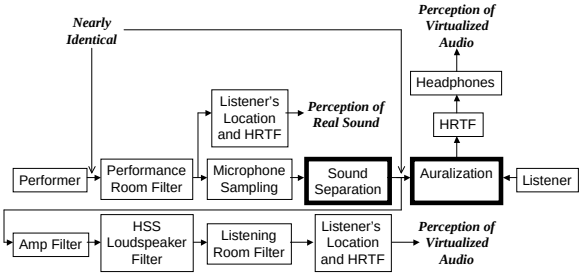


Figure 4: Virtualized audio: filtering, separation, auralization.

popular.

# 3 Virtualized audio

Figure 3 shows what an virtualized audio system would look like, both for the headphone listener who wants to insert himself into a virtual acoustic space (lower right), perhaps that of the performer, and the listener within a physical room (upper right). Figure 4 shows the corresponding filter chain model of virtualized audio. The performance room is unchanged, although it might be desirable to introduce more or different microphones or to arrange them differently. The action is in what happens with the mic feeds. Instead of mixing the signals from these mics down to a small number of output channels, a virtualized audio system instead uses these signals to reconstruct the original signal being produced by the performer (or performers). This process, commonly referred to as source separation, is discussed in more detail in Section 4.

With a separated source signal, it is only necessary to emit that signal in the listening room at the desired position. The room and the listener's head will do the rest. However, current loudspeakers are not able to throw their voices in this way. American Technology Corporation, however, has a nascent technology, HSS (Hypersonic Sound System) that appears to be able to do this. The limitations and costs of this technology are not yet clear to this researcher. Unfortunately, ATC has only limited numbers of samples at the moment. Previous attempts to build speaker systems such as HSS have failed. A phased array (see next section) of standard dynamic loudspeakers is a possibility if a more limited range of performer and listener movement is acceptable.

While the feasibility of the upper right portion of Figure 3 remains to be seen, the lower right portion appears to be completely feasible. To introduce a headphone listener into a virtual room populated by the performer is the bailiwick of auralization. Auralization is discussed in more detail in Section 5. Following the auralization stage is the HRTF, which, in the case of headphones, must be artificially introduced. With respect to Figure 4, it is important to note that auralization requires not only the separated source signal, but also the composition of the virtual room and the location and orientation of the listener within it.

# 4 Reverse problem: separation

In its most general form, source separation is referred to as a blind source localization and blind deconvolution problem. The problem statement is roughly this: given the signals captured by the microphones and the positions of the microphones, find out how many sound sources (performers) there are, where they are located, and, for each source, separate out the effects of the other sources and the filtering effect of the room. By "blind" it is meant that we may assume nothing about parameters such as the number of the sound sources (performers), the nature of the signal emitted by each source, or various properties of the room. The localization portion of the problem involves finding the number of sources and their locations in the room. The deconvolution part of the problem involves reversing the effects of the performance room filter.

In its fully blind form, the problem is essentially a statistical estimation problem. It may be addressed using the expectation maximization algorithm, for example. The problem becomes easier with more microphones. An essential element to any successful solution of the problem is being able to bring significant compute power to bear. Once the number of sources, their locations, and the room filters are known, a filter can be constructed that extracts a particular source from the audio streams the mics produce. This filter remains useful until more sources come on-line, the location of a source changes, or the room changes. If we can simply localize sources, we may be able to reuse work in characterizing the room response. Indeed, one can envision doing a centimeter-by-centimeter response

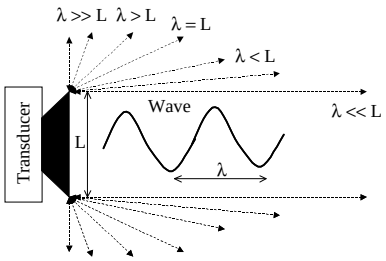


Figure 5: Transducer beaming.

map of the room to each mic and thus avoiding most of the work in the deconvolution step.

We can un-blind the problem in other ways as well. For example, knowing the number of sources and their characteristics (human speech, type of musical instrument, etc) helps considerably. Another example would be tracking the sources via a badge or some kind of visual scheme, in concert with camera tracking, for example.

If we know the location of the performer, we can "zoom in" on him acoustically using an interesting property of transducers that is illustrated in Figure 5. When the wavelength produced by (or intercepted by) a transducer is significantly smaller than the size of the transducer, the transducer acts like a searchlight, beaming sound (or its attention) in a specific direction. It might be practical to exploit this feature of nature by mounting a highly directional microphone on a face-tracking camera system. Surprisingly, mechanical tracking and even highly directional transducers are not really needed. A collection of omnidirectional microphones can be electronically processed to cast a beam of attention in a particular direction, as shown in Figure 6. This is known as a phased array. The selected direction is transformed into a set of phase shifts (delays), one per microphone, that, when the phase-shifted microphone feeds are combined, makes the array of small microphones appear to be a large microphone pointed in the desired direction. It is possible to do the same thing with an array of loudspeakers, beaming sound in a desired direction.

It is important to note that much of the technology behind source separation was developed in the context of submarine sonar and aircraft radar during the cold war. Separation has also been an important subgoal of voice recognition system developers, as well as an area of interest for acoustics researchers. Much of this work is ripe to be leveraged for applications such as the Access Grid. Furthermore, by exploiting the significant computational power available or soon to be available from the Grid, we can potentially solve increasingly blind versions of the problem, replacing expensive custom equipment with computational cycles.

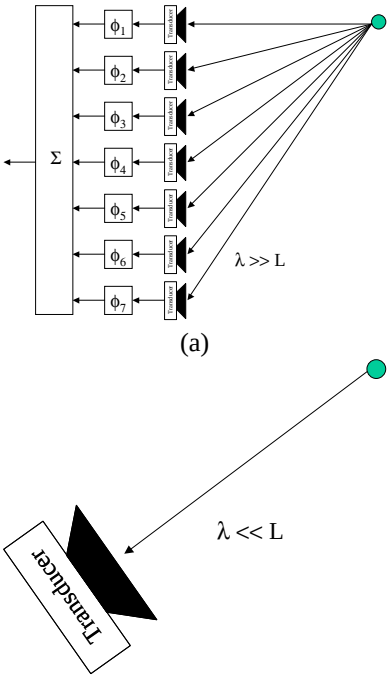


Figure 6: An electronically steerable phased array (a) and its physically steered counterpart (b).

# 5 Forward problem: auralization

Auralization is the process of filtering the recovered signal of the separation step (or a raw sound signal from close-micing and echo-canceling the performer) to impart the sound of a virtual room. The output of the auralization step is a binaural signal that is filtered by an HRTF and then injected into a pair of headphones, giving the listener the impression that he is in the virtual room along with the performer.

Filtering is the simpler part of the auralization process. Determining the filters (there is one for each pair of sound source and listener) is considerably more difficult. Normally, we limit ourselves to LTI (linear and time-invariant) filters. Linearity lets us consider each source independently and then sum the results. Linearity and time-invariance greatly simplifies the filtering process. Usually, the filter is further restricted to be a FIR, IIR, or joint FIR/IIR—a rational function. The linearity assumption is quite reasonable. On the other hand, time-invariance only holds until a sound source or the listener moves, at which time a new filter must be determined. A number of techniques and libraries have been developed to do efficient LTI filtering of audio signals on personal computers. It is important to be able to track the listener's location in the room and the orientation of his head, as the filters depend on this information.

The traditional approach to computing the filter for a

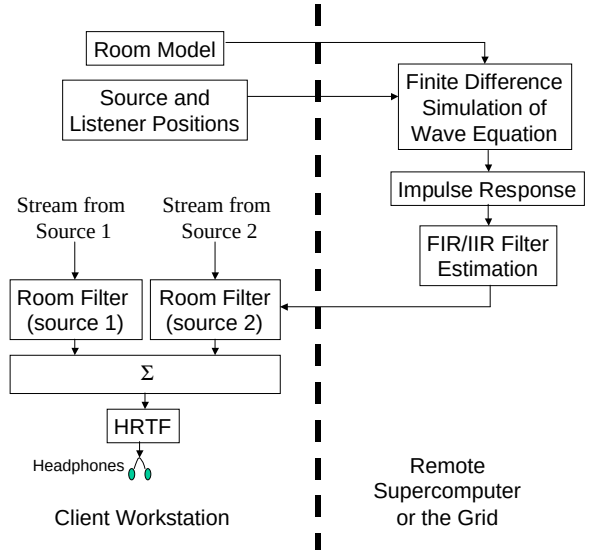


Figure 7: Auralization based on physical simulation of wave equation.

given room is ray-casting. We cast out a ray from the sound source, bounce it against the walls, dissipate it, and wait for it to impinge on the lister position, at which point we capture the time delay and the amplitude. This is computationally intensive process, but almost embarrassingly parallel. If the room geometry is relatively simple, it can be "unfolded" geometrically, making it possible to cast a much smaller number of rays to get the same result.

At this point it is important to discuss how we would go about determining the filter for a physical room. We would place a microphone at the listener position, snap our fingers at the sound source position, and record what the microphone sees. This recording would be the impulse response of the room between those two points. A linear filter is fully characterized by an impulse response, which can be though of as an infinite FIR filter. To make a tractable filter out of the impulse response, we would either truncate it into an FIR or we could estimate the coefficients of an FIR/IIR filter from it.

Given sufficient computational resources, we can do precisely the same thing in a simulated room. A finite difference simulation of the evolution of the wave equation (the initial condition is the finger snap, or injected impulse in pressure) is sufficient for us to compute the impulse response. Figure 7 demonstrates what a system that does this would look like. In steady state, the client workstation would simply apply its room filters to each of the source audio streams, sum the results, apply an appropriate HRTF, and then feed to the user's headphones. When the room changed or a source or listener position changed, the room model and positions would be shipped over to the finite difference simulation. The simulation would reset, inject an impulse at the source position, step forward an appropriate number of times, and collect the values at the listener position. It would then estimate a filter based on this impulse response and ship it back to the client.

Note that the above places interesting requirements on the supercomputer or Grid running the simulation: (1) the invocations of the simulation are at the behest of the user and (2) the simulation must respond in a bounded amount of time. Taken together, these requirements are those of an interactive application, but this is an interactive application that leverages real physics. It is also important to note that this model places minimal requirements on the communications network. To invoke the simulation requires only a simple room model and a set of positions. The simulation only returns a FIR/IIR filter. Even if it returned the whole impulse response, the computation to communication ratio would still be fantastically high. These features make virtualized audio a prime candidate for an interactive Grid application.

The reader may at this point be wondering why we don't simply eliminate the impulse response and filter altogether and simply pump the audio streams right into the finite difference simulation. To do this would require that we be able to step the simulation in real-time. To do so would require that we operate the simulation at a rate of $O(xyz(kf)^4/c^3)$ stencil operations (about 30 floating point operations each) per second, where $xyz$ is the volume of the room, $k$ is the number of grid points per wavelength, $f$ is the maximum frequency to resolve, and $c$ is the speed of sound in the medium. For air, $k = 2$, $f = 20$ KHz, and $x = y = z = 4$ m, this amounts to about $4.110^{12}$ stencil operations per second, or well within the 10s of petaflops. In other words, the impulse response approach is needed to make this form of auralization feasible today.

# 6 Conclusions

This paper has been to introduce virtualized audio, an alternative approach to audio systems that attempts to preserve information and thus is able to replace pathetic illusion with an attempt at reality. We have looked at how the two subproblems of virtualized audio, namely separation and auralization, can be addressed with specialized hardware or by bringing to bear the computational resources that grid computing is likely to offer us.

A prototype of the auralization step of virtualized audio is currently being developed as a student project at Northwestern.