Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2004)
O. Deussen, C. Hansen, D.A. Keim, D. Saupe (Editors)

Auralization I: Vortex Sound Synthesis

Youngin Shin† and Chandrajit Bajaj‡

Department of Computer Sciences
Center for Computer Visualization
University of Texas at Austin
Austin, Texas USA 78712-0233
Institute of Computer and Engineering Sciences
http://www.ices.utexas.edu/

Abstract

Auralization is the process of extracting and displaying meaningful information in the form of sound from data. Through not only visualization but also auralization, users may have better understandings of the data, especially when it is visually complicated. In this work, a field auralization technique is introduced, which objective is at the sound synthesis from field information represented as 3D time-varying volume data. Our technique takes a hybrid approach between parameter mapping and direct simulation. During preprocessing, acoustic strengths are computed at each vertex at each time step of volume data. During interaction, users navigate within the volume space and audio frames are computed by integrating the radiations from the sources. A number of problems inherent in this problem and our solutions are discussed.

Categories and Subject Descriptors (according to ACM CCS): I.3.m [Computer Graphics]: Miscellaneous

1. Introduction

Auralization is the process of extracting and displaying meaningful information in the form of sound. In a physically-based sound synthesis, accurate computation of sound generated from physical phenomena, e.g. the vibration of surfaces at object collisions is of interest, while in simple sonifications, data may be numerically mapped to different acoustical properties such as frequency or amplitude. Here, we introduce a technique called *field auralization*, which objective is at the computation of physically meaningful sounds from field information represented as 3D time-varying volume data. As volume rendering techniques visually display data, our method aims at its acoustic display on top of visualization for enhanced perception through increased modality. Note that we compute physically meaningful sounds that are different from real sounds just as the color values of result images from volume renderers are different from the real colors.

In our simulation, we used a 3D time-varying scalar field data that contains density, temperature, and pressure values obtained by discretely sampling the contributions of SPH particles including dark matter at each grid point during cosmological explosions. When visually rendered, one can see gas and clouds from cosmological explosions. Naturally, some of them may be obscured by others sitting before them, thereby making it difficult to perceive phenomena occurring behind, e.g. a minor explosion or turbulence. However, if appropriate sounds are generated from them, it can travel through and reach the listener, increasing the chance of being noticed.

To generate such sounds, given data is first mapped from its domain onto acoustic properties. Sound sources are then detected from which the radiations are integrated to compute sounds when the listener position is known. In our work, density values in cosmic space among the fields in the given data are mapped onto the density and the particle velocity in air. This is because we were interested in generating sounds from turbulence of density. Note that our technique requires the mapping to be done

† codeguru@cs.utexas.edu
‡ bajaj@cs.utexas.edu

onto density and particle velocity values in air but does not have any restriction on what are mapped onto the values as well as how it is done as it is totally dependent on the application. From these acoustic properties, three different types of sound sources, i.e. monopole, dipole, and quadrupole type sources are detected at each time step. As users navigate within the space, the listener position is known and the radiations from the sources that arrived there at that time are summed to obtained the sound pressure.

A short review of auralization steps in general is given to address problems we face in auralizing field data. Auralization is typically composed of three major steps: *synthesis*, *rendering*, and *localization*. During the first step, *synthesis*, sounds generating from sources are computed. Possible sources of sounds include object surface vibrations and turbulent flows such as aircraft jet engine where there is no vibrating boundaries or objects that generate disturbance in the medium. Once generated, sounds radiate into the medium as acoustic waves. During the propagation, sound experiences various physical phenomena such as reflection, refraction, attenuation due to absorption, and diffraction. Therefore, sound heard by users should be quite different from that generated by sound sources, depending on many factors including the distance sound traveled, objects that sound reflected or transferred, the type of the medium, etc. During the second step, *rendering*, we compute sound heard by the listener, taking all of these effects into account. The last step is *localization*. Until upto this step, sound is calculated for users in virtual environments, which is independent of the actual configuration of speakers in the real environment. However, real world speakers generally can't exactly simulate the sounds in virtual environment because they are not exactly located at sound sources in virtual environment. Therefore we need to map virtual sounds to a distribution of audio signals for real world speakers to give the best approximated spatial aural images to the audience. This has been a long challenging problem in acoustics because multiple loudspeakers do interfere with each other and this makes the process of analyzing their collective effect difficult.

Considering above steps in the auralization of field data yields a number of problems inherent in this work. First is *acoustic mapping* which is to find good mappings between field values in the given volume data to acoustic properties such as pressure, density, and particle velocity. This is necessary when given data and sounds to be produced are in different domains. Second is *interpolation* which is required if the volume sampling rate and the audio sampling rate are different. The sampling rate of typical volume data is 30 frames per second or, more generally, independent of the playback rate and the data usually contains around several hundreds to thousands of time frames. Therefore, direct play at 44.1 KHz of the data will last only for a very short period of time and be incorrect. If played at 30 Hz at the same rate as animations, we instead can barely hear anything as most audible components are lost. Therefore, interpolation and supersampling is a natural approach to resolve this difference between the two rates. The key problem in our work is *pressure computation* at a given time and location in space, which is an auralization method referred to as *direct simulations* [KM94]. Our method of pressure computation initiated from the idea that turbulence in a media where no vibrating objects or objects exist forms a region of quadrupole sources. This was first pointed out by Lighthill and an accurate analytical solution of sounds from such phenomena was given in [Lig52]. However, this solution is not directly applicable to our problem. It requires an integration of radiations of infinitesimal quadrupole sources distributed around source region. But we are not given any geometrical information of the source region. Therefore, evaluating the integral solution to compute the pressure is not trivial. Our method thus proposes a simple algorithm of source detection and source region computation needed in this evaluation process. Note that it is impossible to obtain accurate continuous solutions because we are given discrete data but also that it is not necessary because our objective is not at the physically accurate simulation but at auralization.

The work of this paper is done as part of an on-going project of building an immersive audiovisual scientific simulation environment that combines all of the three steps of auralization. We mainly focus on the problem of *pressure computation* and present our sound synthesis algorithm and system framework in this paper. The experiments are performed in the VisLab studio at ACES building, UT Austin to take advantage of screens and speakers that surround the center space which forms an excellent 3D immersive environment for our scientific simulation. See Figure 1 for the configuration. The eight boxes are the speakers directed to the center of the room where the user is standing and in the front and back, are a hemi-cylindrically shaped screen and five flat creens.

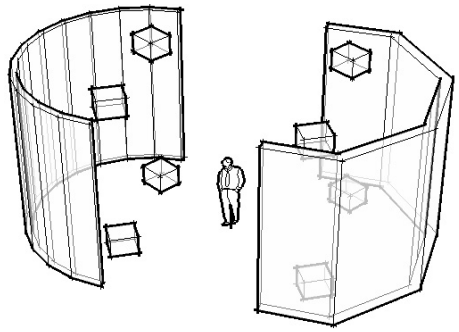


Figure 1: *UT VisLab Environment*

In an environment where a set of loudspeakers are used, care must be taken to take into account the effect of their

interactions as well as those with other objects when accurate spatialization is required. In UT VisLab, there are many sources of indirect sounds, e.g. floor, screens, ceiling, and projectors. In this work, however, we focus only on sound synthesis and problems related to rendering and localization including indirect sounds are not discussed.

The paper is structured as follows. A brief review on related work is given in section 2. Our synthesis technique including a detailed discussion on the physics of sound generation and their mathematical solutions, preprocessing volume data for source detection is described in section 3. Our system framework is presented in section 4, followed by discussion about the result of applying our technique to time-varying data and future work.

2. Related Work

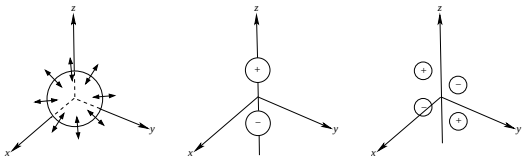
Little research into physically-based sound synthesis from field data has been published. In [KM94], two examples of auditory display are presented, one in parameter mapping and the other in direct simulation. In the first, the simulation of an artificial heart pump produced locations of blood cells, as well as pressure and vorticity at discrete points in the chamber. The total pressure and vorticity at each time step were then mapped to MIDI sounds. In particular, this system could notify users of global changes in the system including opening and closing of heart valves and overall pressure changes on the pusher plate. However, parameter mapping technique is useful when it is feasible to find nice mappings from given data to audio parameters. In the second, direct simulation technique was used in the analysis of data from a tonal acoustic simulation done on rotor-stator interaction inside a jet turbine. This required a flow solver produce time-dependent solutions to obtain pressure samples over time. In [Chi01], Childs summarized the benefit of using sonification in the analysis of Computational Fluid Dynamics (CFD) data; the recognition of significant patterns in the (sonification) data which are not apparent from visual displays, increased productivity in examining large amounts of data by concurrent visual and aural displays, the sonic codification of global events in the flow field which are difficult to perceive from the visual display of local details, and the comparison of sound rendered from simulations of the pressure field with recorded sounds from the corresponding experiments. CFD software was used to solve fluid flow and produced the velocity and pressure values that were mapped to a number of audio parameters including pitch, envelope, duration, and timber. More examples of using direct simulation can be found in [VDDP96, JFOE01, JFOG02]. However, direct simulation is not easily applicable when necessary data for flow solvers, e.g. geometry of the region of interest, or boundary conditions, etc., are not available. Our technique takes a hybrid approach between parameter mapping and direct simulation. First, given field data is mapped onto acoustic parameters, e.g. density and particle velocity rather than audio parameters such as pitch, amplitude, or timber. This facilitates a physically-based sound synthesis based on the control of sound source strengths using acoustic parameters. Second, radiations from turbulence are computed by evaluating the Lighthill's solution rather than using flow solvers, which is particularly useful when only time-varying volume data is provided.

3. Synthesis

3.1. Physics of Sound Generation

3.1.1. Sound Sources

Sound is the disturbance of pressure of the medium with frequency components within an audible frequency range, i.e. 20Hz $\sim$ 20,050Hz. The sources of such disturbance, i.e. sound sources are mainly of three types, i.e. monopoles, dipoles, and quadrupoles, which is shown in Figure 2.



(a) monopole (b) dipole (c) quadrupole

Figure 2: *Three types of sound sources*

A *monopole* is a volume point source modeled as a pulsating sphere. Usually, the strength of a monopole is given as the rate of increase of fluid volume per unit volume of the fluid. A *dipole* is a pulsating volume source with two monopoles of equal strength but opposite signs placed a short distance apart, whose net volume source strength is zero. A *point* dipole is a dipole where the two monopoles are so close such that it is equivalent to a point volume source that has a strength given as the divergence of a time-dependent vector. A *quadrupole* is a volume source involving two space derivatives, which is equivalent to a combination of four monopole sources, whose net volume source strength is zero.

Consider a wave equation,

$$\left(\frac{1}{c_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) p = \mathcal{F}(\vec{x},t), \tag{1}$$

where a generalized source $\mathcal{F}(\vec{x},t)$ generates the wave radiating away from the source region. Using the free-space Green's function, we obtain the solution of this equation as

$$p(\vec{x},t) = \frac{1}{4\pi}\int_{-\infty}^{\infty} \frac{\mathcal{F}\left(\vec{y}, t - \frac{|\vec{x}-\vec{y}|}{c_0}\right)}{|\vec{x}-\vec{y}|} d^3\vec{y}, \tag{2}$$

which is called a *retarded potential*. See [Bla98, How02] for a more detailed discussion. If we assume that $\mathcal{F} = 0$ outside of the source region V, then we can restrict the integral to V only.

First, the strength of a monopole, $q(\vec{x},t)$, is given as the rate of increase of fluid volume per unit volume of the fluid and we have $\mathcal{F}(\vec{x},t) = \rho_0 q_t(\vec{x},t)$, where ρ_0 is the ambient density of the medium, and q_t is the time derivative of q. However, we will convert $\rho_0 q_t(\vec{x},t)$ into a form described in terms of mass flow, i.e. the rate of increase of mass per unit volume (or per voxel divided by the volume of a voxel),

$$\rho_0 q_t = \rho_0 \frac{1}{V_c}\frac{\partial V}{\partial t} = \frac{1}{V_c}\frac{\partial(\rho_0 V)}{\partial t} = \frac{1}{V_c}\frac{\partial M}{\partial t}, \tag{3}$$

where V is the fluid volume per voxel, V_c is the volume of a voxel, and M is the mass in a voxel. We further convert this so that the mass flow is written in terms of density fluctuation. For this, we assume that the mass in a voxel may be approximated by

$$\frac{\partial M}{\partial t} = V_c \frac{\partial \tilde{\rho}}{\partial t}, \tag{4}$$

where $\tilde{\rho}$ is the density at the a point inside the volume source region. From this, we can rewrite $\rho_0 q_t$ as

$$\rho_0 q_t = \frac{1}{V_c}\frac{\partial M}{\partial t} = \frac{1}{V_c} V_c \frac{\partial \tilde{\rho}}{\partial t} = \frac{\partial \tilde{\rho}}{\partial t}. \tag{5}$$

The solution is then

$$p(\vec{x},t) = \frac{1}{4\pi}\int_V \frac{\partial \tilde{\rho}}{\partial t}\frac{1}{|\vec{x}-\vec{y}|} d^3\vec{y}. \tag{6}$$

For a dipole source, the strength is given as a divergence of a vector quantity $\vec{f}(\vec{x},t)$. Physically speaking, two almost cancelling monopoles represent a dipole. This means that the net mass flow is zero, while the momentum flow isn't and it is this momentum flow that emits sounds. In this sense, we compute the expression for $\vec{f}$ as the *momentum flow* per unit volume, i.e. the rate of increase of momentum per unit volume (or per voxel divided by the volume of voxel). Mathematically stated, it becomes

$$\vec{f} = \frac{1}{V_c}\frac{\partial \vec{m}}{\partial t} = \frac{1}{V_c}\frac{\partial(M\vec{v})}{\partial t} = \frac{1}{V_c}\frac{\partial(\tilde{\rho} V_c \vec{v})}{\partial t} = \frac{\partial(\tilde{\rho}\vec{v})}{\partial t}. \tag{7}$$

where $\vec{v}$ is the velocity at a point inside the volume source region and $\vec{m}$ is the momentum flow. Then, we have

$$\mathcal{F} = \nabla \cdot \vec{f} = \nabla \cdot \frac{\tilde{\rho}\vec{v}}{\partial t}. \tag{8}$$

and applying this to (2) yields

$$p(\vec{x},t) = \frac{1}{4\pi}\int_V \frac{\partial}{\partial x_i}\left(\frac{\partial(\tilde{\rho} v_i)}{\partial t}\right)\frac{1}{|\vec{x}-\vec{y}|} d^3\vec{y}. \tag{9}$$

For a quadrupole source, the source is specified as a tensor T_{ij}. If we assume that the source flow is incompressible, which is often possible when the characteristic Mach number $M \sim v/c_0$ is small (specifically, when $M^2 \ll 1$), acoustic effects in the source flow including self-modulation, convection, refraction, and attenuation are negligible [How02]. In an unbounded turbulent flow, this tensor quantity, called Lighthill's tensor is given as

$$T_{ij} = \rho v_i v_j + \Big((p-p_0) - c_0^2(\rho-\rho_0)\Big)\delta_{ij} - \sigma_{ij}, \tag{10}$$

where δ_{ij} is the Kronecker delta function and σ_{ij} is the viscous stress tensor for a Stokesian fluid. Under the assumption that the source flow is incompressible, $T_{ij} \approx \rho_0 v_i v_j \approx \tilde{\rho} v_i v_j$. The strength of a quadrupole is then given as

$$\mathcal{F}(\vec{x},t) = \frac{\partial^2 T_{ij}}{\partial x_i \partial x_j} \approx \frac{\partial^2(\tilde{\rho} v_i v_j)}{\partial x_i \partial x_j}. \tag{11}$$

Substituting (11) into (2) yields

$$p(\vec{x},t) \approx \frac{1}{4\pi}\int_V \frac{\partial^2(\tilde{\rho} v_i v_j)}{\partial x_i \partial x_j}\frac{1}{|\vec{x}-\vec{y}|} dV, \tag{12}$$

3.2. Sound Synthesis

3.2.1. Introduction

Our sound synthesis technique is composed of two steps: preprocessing and interaction. During preprocessing, acoustic mapping and source strength computation are performed. In acoustic mapping, the density and the gradient of density in the given SPH field data are mapped onto density and particle velocity in air. At every vertex, three different types of source strengths are computed, which are used in later examinations whether they are sound sources or not based on our criteria explained in the following section. During interaction, user locations and orientations are known as a user navigates in the space. Once these information is obtained, we can compute audio frames as pressure values at our desired audio sampling rate by identifying all the sources whose radiations can affect pressure at the user location at any given time and taking their summation.

```
SYNTHESIZE-SOUND(L, t)
begin
  τ ← t · SR_v / SR_a
  p ← 0
  for all types of sources at time interval τ
  begin
    S ← FIND-REACHABLE-SOURCES(L, t, τ)
    for sound source s in S
    begin
        p ← p+ COMPUTE-PRESSURE(L, t, τ)
    end
    audio-frame[t] ← p
  end
end
```

Above shows our sound synthesis algorithm. $\vec{L}$ is user location vector, t is the elapsed time from the beginning of the simulation given as an integer starting from 0 that increments at every $1/S$ seconds, where S is the audio sampling rate, and SR_a and SR_v are the audio and volume sampling rates, respectively. τ is the time step index of the volume data where t falls into. In the following sections, each step of this algorithm is discussed.

Note that the user movement during navigation is first captured and then the audio frames are computed. Alternatively, one can do a real-time computation of audio frames during user navigation based on the same idea by precomputing pressure functions as well at each vertex. Current approach is not real-time as it takes long time to compute audio frames but requires less time and memory than the alternative approach in preprocessing as source strengths but pressure functions are computed. We did not take the real-time approach for now as our interest was in the validity of our sound synthesis technique rather than performance in time.

3.2.2. Acoustic Mapping

The problem of acoustic mapping is to find good mappings between field values in the given volume data to the acoustic properties for further sound processing. The major physical properties in acoustics are the sound pressure p, the excess density $\delta\rho$, and the particle velocity $\vec{u}$. Acoustic mapping plays an important role as the knob of adjusting the quality of the sound generated. The transfer function that maps the field values in the volume data to the acoustic properties depends heavily upon the characteristics of the data. In one application, the electrical charge around an atom may be directly mapped to the density, while the changes in the density in space simulating cosmological explosions may be mapped to the particle velocity. In our work, we are given an cosmological explosion volume data with field values including density and temperature represented as logarithm of base 10 at each grid point. The density values were around the range $[-2.2,-0.5]$ and we took power of 10 of the density scaled it so that the values fall in the rage $[0.5,15]$ which is used as the density value ρ in air. The particle velocity $\vec{v}$ or v_i with subscript $i = 1,2,3$, is then obtained from taking $\nabla\rho$ and scaling it so that v_i's fall in the range $[-150,150]$.

3.2.3. Source Strengths Computation

As discussed in 3.1, $p(\vec{x},t)$ is computed by the integration of the radiations from sound sources distributed over the source region V. To find V from the given time-varying volume data, we detect acoustically strong vertices as sound sources, which are called *source points*. As there are three different types of sound sources, we compute three different types of source strengths as well at each vertex at each time step. In our method, a vertex is detected as a source point if its source strength is higher than the given threshold value. The strength for each type of sources at a vertex is computed as follows. First, for monopole strength, the fluctuation of mass inflow, i.e. $|\partial\tilde{\rho}/\partial t|$ is computed. With current acoustic mapping, this value falls roughly within $[-5000,5000]$. If this quantity is higher than a certain threshold value, we treat this vertex as a monopole source point. Second, for dipole strength, the fluctuation of momentum inflow, i.e. $|\nabla\cdot[\partial(\tilde{\rho}\vec{v})/\partial t]|$ is computed. Rewriting the expression with subscript i, we have

$$\nabla\cdot\frac{\partial(\tilde{\rho}\vec{v})}{\partial t}=\frac{\partial}{\partial x_i}\left[\frac{\partial(\tilde{\rho}v_i)}{\partial t}\right], \tag{13}$$

where repeated index in a term indicates the summation over $i = 1,2$, and 3. With current acoustic mapping, this value falls within $[-400000,400000]$ with about 60% of them within $[-1000,1000]$. Note that large values here does not necessarily mean that it is a dipole. To be a dipole, the introduction of mass must be almost zero at that point. If this quantity is greater than a certain threshold value and at the same time, the monopole strength was almost zero, then we treat this vertex as a dipole source point. Third, for a quadrupole strength, the convection of momentum, $|\partial^2(\tilde{\rho}v_iv_j)/\partial x_i\partial x_j|$ is computed. With current acoustic mapping, this value falls roughly within $[-4000000,4000000]$ with about 60% of them within $[-100,100]$. Similarly, large values does not necessarily mean that it is a quadrupole. To be a quadrupole, the introduction of mass and momentum must be almost zero at that point. If this quantity is greater than a certain threshold value and at the same time, the monopole and the dipole strengths were almost zero, then we treat this vertex as a quadrupole source point. Rewriting this expression with with subscripts 1, 2, and 3 yields

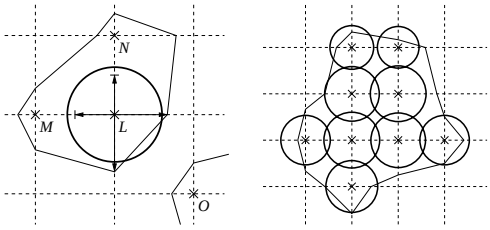
$$\frac{\partial^2(\tilde{\rho}v_iv_j)}{\partial x_i\partial x_j}=\frac{\partial^2(\tilde{\rho}v_1^2)}{\partial x_1^2}+\frac{\partial^2(\tilde{\rho}v_1v_2)}{\partial x_1\partial x_2}+\cdots+\frac{\partial^2(\tilde{\rho}v_3^2)}{\partial x_3^2}. \tag{14}$$

The second derivative in space can be easily obtained by applying twice the same method used in detecting dipole

source points in the desired directions. Note that any source point can be only one of the three different types in our method.

3.2.4. Evaluation of $p(\vec{x},t)$

3.2.4.1. Computing ΔV To compute the pressure, we need to evaluate the integrals in the solutions of $p(\vec{x},t)$ for each type of sources. Theoretically, the source strength must be multiplied by an infinitesimal volume dV and integrated over the source region V to obtain accurate solutions. However, this is not feasible as we are volume data discretely sampled in time and space. Instead, the source region is divided into a partition of *finite* volume ΔV, such that each ΔV contains one source point of strength $\mathcal{F}$. The pressure is assumed to be almost uniform within ΔV, so that the integrals can be approximated by the summation of all $\mathcal{F}\cdot\Delta V$. If ΔV is small, then this assumption is admissible. There may be many different ways of defining and partitioning the source volume and evaluating the integral depending on the applications and the level of required accuracy. For example, one could take an isocontour that encloses the source points and partition it using Voronoi diagram such that each of the partitioned space contains a source point. In our method, source regions are modelled as a set of spheres each of which is centered at a source point because of its simplicity in implementation while providing enough accuracy for our application. See Figure 3 for example.



(a) Cut Distances (b) ΔV Spheres

Figure 3: *Computing ΔV*

where L, M, N, and O are source points marked as $\times$ in Figure 3 (a). The thin solid polygon at center is the isosurface taken at the threshold value of source strength for a certain type of source which encloses source points L, M, and N. Consider the circle centered at L that denotes the sphere representing the ΔV for L. Among the direct neighbours of L, only M and N are source points. Therefore, isosurface will cut edges connected to L except those connecting to M and N. For edges that are cut by an isosurface, the cut distance is the length from L to the cut point. For edges that connect to direct neighbours that are not cut by an isosurface, the middle point is the cut distance. The arrows starting from L in Figure 3 (a) indicates its cut distances. In 3D space, there are at most six direct neighbours and the average of all the cut distances is taken as the radius of the sphere. Figure 3 (b) shows a set of ΔV spheres for the given set of source points.

3.2.4.2. Finding Reachable Sources Since the distance from one source point to the listener is different from that from another source point, radiations from some source points may have reached or passed the listener while those from the others are not reached yet. See Figure 4 for example. The listener moves along the dashed line and is currently at location L at time t_0. At the same time, the radiations from sources s_1, s_2 and s_3 have propagated omnidirectionally upto the distance plotted as circles around the source points. As shown in Figure 4, the radiation from s_1 has not been propagated to location L, while those from the other two source points did. Therefore, we only add the radiations from only s_2 and s_3 for the pressure value at time t_0.

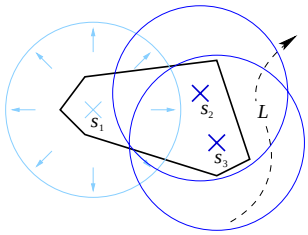


Figure 4: *Listener and Volume Source*

3.2.4.3. Interpolation of $\tilde{\rho}$ and v_i To find reachable sources as discussed above, we need to determine if a vertex is a sound source at any time. Therefore, we used interpolations over the series of discretely sampled strength values at that vertex over time. A linear interpolation is used for speed but other higher order interpolation algorithms may be used to avoid sharp changes at each time or sample point if expensive computation is tolerable. In this work, however, no formal study is done on the effectiveness of using higher order interpolations in terms of sound quality.

3.2.4.4. Computing $p(\vec{x},t)$ We approximate the integral expression introduced in 3.1 for each type of sources by dividing the source region into a partition of finite volumes and adding the radiations from the reachable subset of source points. As discussed in 3.2.4.1, the integral appearing in the expression of $p(\vec{x},t)$ from 3.1 is evaluated as the summation of $\mathcal{F}\cdot\Delta V$. The source strength $\mathcal{F}$ is computed from $\tilde{\rho}$ and v_i, which are obtained using cubic interpolation as discussed in 3.2.4.3. The expressions of $p(\vec{x},t)$ for monopole, dipole, and quadrupole sources are

$$p(\vec{x},t) = \frac{1}{4\pi}\sum_{k=1}^{K_m}\left[\frac{\partial\tilde{\rho}_k}{\partial t}\frac{\Delta V_k}{D_k}\right], \tag{15}$$

$$p(\vec{x},t) = \frac{1}{4\pi}\sum_{k=1}^{K_d}\sum_{i=1}^{3}\left[\frac{\partial^2(\tilde{\rho}_k v_i^k)}{\partial x_i \partial t}\frac{\Delta V_k}{D_k}\right], \text{and} \tag{16}$$

$$p(\vec{x},t) = \frac{1}{4\pi}\sum_{k=1}^{K_q}\sum_{i=1}^{3}\sum_{j=1}^{3}\left[\frac{\partial^2(\tilde{\rho}_k v_i^k v_j^k)}{\partial x_i \partial x_j}\frac{\Delta V_k}{D_k}\right], \quad (17)$$

where K_m, K_d, and K_q are the numbers of reachable monopole, dipole, and quadrupole type source points, ΔV_k is the finite volume of source point k, D_k is the distance from the user location to source point k, and $\tilde{\rho}_k$ and v_i^k are the average density and particle velocity in i direction at source point k.

4. Implementation and Results

4.1. Implementation

Our system is implemented for SGI Onyx2 machine with an 8-channel ADAT output which are connected to eight loudspeakers mounted in VisLab studio at ACES building, UT Austin. See Figure 1. The user stands in the middle of the space facing the center of the hemispherical screen at front. First, the field data is loaded and preprocessed. Then, as user navigates in the space using the mouse, the location and the orientation are captured. Once captured, our synthesizer computes the audio frames at a specified audio sampling rate as follows. However, due to the difference between the sampling rate of mouse positions and the audio sampling rate, we do a linear interpolation to obtain user locations between mouse samples. See Figure 5 for example. The dotted curve shows the movement of the user position L and $\times$ indicates where the location and the orientation are captured. In our system, capture occurrs at around every 0.17 second, i.e. 6 $\sim$ 10 Hz, with a variation of 0.01 second and the audio sampling rate is 44.1 KHz. Consider the two captures at location L_1 and L_2 at time t_1 and t_2, respectively, shown in Figure 5. From the difference in time, we know that there are $\Delta t \cdot 44100$ audio frames to sample in this interval, where $\Delta t = t_2 - t_1$. The solid line shows our linear interpolation of user location and orientation and at some time t, where $t_1 \le t \le t_2$, user location is computed as L as well as the orientation. The user orientation is not used in our synthesis process though it is captured. This is for our extending work of localization, which is not discussed in this paper. The synthesizer computes $\Delta t \cdot 44100$ audio frames by computing pressure values at the interpolated location L at time $t_1 \le t \le t_2$.

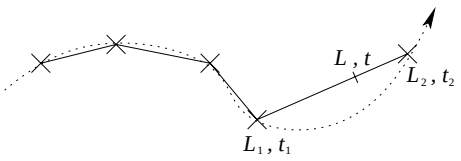


Figure 5: *Capturing User Movement*

4.2. Results

At the start of the simulation, all dark matters are at the center of space and they explode, i.e. the particles burst out to the space. Therefore, there is a very strong acoustic radiation from the center of the space with minor local radiations. The ratio of virtual world and the real world in metric unit was 1.0, i.e. distance of unit 1 in virtual space corresponded to 1 meter. For $64\times64\times64$ data, which we used in this work, the longest distance within this space from center point is about 55.43m, which takes about 0.162 seconds for the sound wave to travel assuming the speed of sound $c_0 = 343$m/s. In our simulation, one can hear new sounds as sources are formed within audible distance. The strength of the sources affects the loudness of the sound, while turbulence or vorticity, mainly caused by quadrupole sources, produces high frequency components. Also, users can detect phenomena that are occluded or out of the field of view through the introduction of new frequency components.

We conclude this section with a discussion of efficiency with respect to the time taken for preprocessing and computing single audio frame after preprocessing. Following table shows the average time taken for preprocessing each time step for each type of sources with varying dimensions of the data. The numbers are time measured in seconds.

Dimension	Monopole	Dipole	Quadrupole
$10\times10\times10$	0.064	0.056	0.065
$20\times20\times20$	4.001	3.912	4.074
$30\times30\times30$	44.990	42.735	46.812
$40\times40\times40$	261.664	254.755	264.676
$50\times50\times50$	1011.354	1002.721	1012.081
$64\times64\times64$	6201.703	6183.029	6210.341

As the size of dimension in each direction increases linearly, the algorithm seriously slows down as the order of our source detection algorithm is $O(n^3)$, where n is the dimension in each direction. For a large size data such as $512\times512\times512$, this algorithm will take longer than a feasible duration. One can improve this by taking the characteristics inherent in the data, e.g. coherence and locality, and coming up with a more efficient algorithm. The next table shows the average time $\tilde{t}$ taken for computing single audio frame with varying the dimensions of the data. The numbers are time measured in microseconds.

Dimension	$\tilde{t}$
$10\times10\times10$	7
$20\times20\times20$	16
$30\times30\times30$	23
$40\times40\times40$	65
$50\times50\times50$	102
$64\times64\times64$	236

As the size of dimension in each direction increases linearly, $\tilde{t}$ increases almost linearly. Since $\tilde{t}$ depends heavily upon the

number of detected source points, it increases as the sampling resolution increases.

5. Conclusions

Our work proposes a novel idea of synthesizing sounds from time-varying field data by detecting three different types of sound sources data and evaluating Lighthill's solution over the source region. This technique is useful when direct simulation by solving PDEs is not feasible as it doesn't require geometric representation or boundary conditions. Two problems to be solved are worth making remarks. First, the generated sound during our system is not yet pleasant enough to hear though it seems to be a nice starting point of being a good helper of aurally understanding the data. The problem of generating pleasant sound may be well treated by AM/FM modulation or empirically finding nice acoustic mappings based on the characteristics of give data and applications. Second, this system is slow at preprocessing and sound synthesis during interaction.

6. Future Work

Currently, it is assumed that there are no solid boundaries in the space which may be sources of indirect sounds. It is apparent that indirect sound will be a valueable auditory queue for enhanced data perception as it acoustically reflects the surrounding environment. Our work is to be extended to add sound localization for an immersive scientific syntheic environment as well. Also, we need to work on generating more pleasant sound and making the system more efficient in preprocessing and real-time sound synthesis.

References

[Bla98] BLACKSTOCK D. T.: *Physical Acoustics*. Wiley Intersicence, 1998.

[Chi01] CHILDS W.: The sonification of numerical fluid flow simulations. In *Proceedings of the 2001 International Conference on Auditory Display, Espoo, Finland* (2001).

[How02] HOWE M. S.: *Theory of Vortex Sound*. Cambridge Press, 2002.

[JFOE01] JAMES F. OBRIEN P. R. C., ESSL G.: Synthesizing sounds from physically based motion. In *Proceedings of SIGGRAPH 2001* (2001), SIGGRAPH.

[JFOG02] JAMES F. OBRIEN C. S., GATCHALIAN C. M.: Synthesizing sounds from rigid-body simulations. In *Proceedings of SIGGRAPH 2002* (2002), SIGGRAPH.

[KM94] K. MCCABE A.: *Auditory Display of Computational Fluid Dynamics Data*. Addison Wesley, 1994.

[Lig52] LIGHTHILL M. J.: On sound generated aerodynamically. part i: General theory. In *Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences* (1952), pp. A211:564–587.

[VDDP96] VAN DEN DOEL K., PAI D. K.: Synthesis of shape dependent sounds with physical modeling. In *Proceedings of the International Conference on Auditory Display(ICAD)* (1996), ICAD.