Proceedings of the Institute of Acoustics

THE REAL-TIME SIMULATION OF THE ACOUSTICS OF VIRTUAL ENVIRONMENTS ON PERSONAL COMPUTERS

I Drumm University of Salford, The Department of Acoustics and Electronic Engineering, Manchester, U.K.

1. INTRODUCTION

The paper describes the current development of an interactive real-time acoustic prediction application to teach the acoustics effects of modifications to indoor environments. The aim is to provide a virtual environment in which the student can interact with a virtual lecturer, modify architecture and position furniture, sound sources and acoustics modifiers so gaining an appreciation room acoustics. The paper will discuss techniques considered and used, highlighting the difficulties associated with real time prediction.

In creating a virtual environment the application has to model a three dimensional graphic representation of the space and objects within. Previously the preserve of military simulations and dedicated high end computer hardware, high quality computer graphics applications for real time simulation of virtual environments are now commonplace on everyday personal computers. This has been facilitated by

- The huge demand for 3D applications for Design, Simulation, Data Visualisation and Games.
- The appearance of popular standard application programming interfaces (APIs) such as OpenGL (Segal & Akeley[1]) and Direct3D (Microsoft[2]) providing relatively easy to use yet fast and effective functionality for the rendering of 3D graphics.
- Cheap yet very effective 3D accelerating hardware that optimise much of OpenGL's and Direct3D's functionality. The latest generation of graphics cards based on Voodoo III, TNT2 and Savage4 chipsets offer near video quality graphics for prices around £100.

The effective real-time simulation of the acoustics of virtual environments has until recently proved more difficult due to the large demands on the CPU bandwidth. However as with graphics, new acoustic APIs and cheap accelerating hardware for acoustic rendering are beginning to appear, notably the A3D API (Aureal[3]) and accompanying Vortex 2 PCI audio processor.

2. SIMULATING AUDIO

2.1 Sound Propagation

As sound propagates from a source in all directions, much of it will be specularly and diffusely reflected or occluded by walls and objects before reaching the listener. A large proportion of the original sound energy will be lost to wall and air absorption and the remainder reach the listener across a range times. This will to be perceived as echoes and reverberation. There are numerous recognised ways to model this sound propagation, notably

- Ray tracing – where the propagation of the sound wave front is represented by an omni directional spread of rays (Kulowski[4]). Each ray can represent a proportion of the initial sound energy reflected around the room. This ray may or may not be incident on a spherical detector representation of the listener. The algorithm automatically takes into account occluding planes though does have inherent systematic errors (Lehnert[5]) where valid ray paths may be missed as rays diverge or invalid rays be incident on the spherical detector.

- Image method – where possible reflected images of sources relative to the listener are calculated for all combinations of walls to specified orders of reflection (Borish[6]). Invalid image combinations are ruled out with visibility tests. This method produces a very accurate impulse response for a given room though demands on computation time increase exponentially with higher orders of reflection or a higher number of reflecting surfaces, hence the need for optimisations (Lee & Lee[7]). Many acoustic prediction applications use the image method in conjunction with ray tracing in a hybrid model (M. Volander[8]).

- Beam Tracing – rather than using rays, the propagating sound can be represented by conical beams (Maercke & Martin[9]) or triangular beams (Lewers[10]). The use of beams can improve speed and reduce errors as an initial high density of beams isn't necessary, a point source can be used and occluding surfaces are taken into account during the process. Unlike previous methods, the calculation of diffuse sound propagation needn't rely on statistical assumptions as the surface areas of beams incident on reflective surfaces can be used as input data for a radiant exchange model. Gaps and duplications in the propagating sound still occur hence the recent development of the adaptive beam tracing algorithm (Drumm[11]). Here the cross-sectional shapes of reflected child beams are determined by the areas of illumination of parent beams incident on walls as shown in Figure1.

All these methods require a large CPU bandwidth when implemented in software alone. Even with sacrificing accuracy and orders of reflection for speed, real-time simulation of sound propagation is beyond the scope of most PCs especially if there are multiple sources and the simultaneous simulation of 3D graphics. Although it is worth noting that there are recent acoustic prediction methods that implement real-time and pseudo real-time prediction via the use of prior calculations integrated with on the fly prediction and dedicated hardware (Hogdson, Heerema, Halingten[12]) (Funkhouser[13]).

2.2 Localisation

Human beings can localise sound sources relative to themselves by virtue of the way sound diffracts around their head and two ears. The resultant differences in frequency and phase of sounds incident on ears helps us guess where sounds are in a way analogous to stereoscopic vision for judging distances. This can be simulated with stereo headphones by convoluting sounds being sent to each ear via head related transfer functions (Kutruff[14]) (Rainer Feistel[15]). These functions use Fast Fourier Transforms to separate the sounds into their frequency components for respective volume and phase adjustments. This requires a high CPU bandwidth and so is difficult to implement in software for real-time applications.

2.3 Application Programming Interfaces

Following in the wake of popular 3D graphics cards there has recently started to appear audio equivalents. Unlike previous sound cards, these have hardware accelerated functionality for simulating room acoustics and localisation. Most simulate room acoustics by letting the developer select from range of generic reverberant effects, however sound cards that use the Aureal A3D API predict sound propagation using a variant of the image method called Wave-Tracing. Developed in conjunction with NASA, Matsushita and Disney the algorithm 'parses the geometry of a 3D space to trace sound waves in real-time' (Aureal[3]).

First order reflections, occlusions and reverberent tails are predicted for up to sixteen sources simultaneously together with the application of generic head related transfer functions to simulate localisation. All this is in real time via accelerating hardware leaving free CPU bandwidth for other processing such as setting up 3D graphics, windows messaging, etc. This functionality is controlled by the Aureal application programming interface (A3D 2.0). This Aureal A3D API mirrors OpenGL's methodology for instance both use matrix transformations to render objects relative to the observer or listener.

In creating a virtual environment all objects, walls etc can be represented as polygons. The visual appearance of a polygon can be given by associated texture (a bitmap image), reflectivity, luminescence and transparency. Similarly, the acoustic effect of the polygon can be described in terms of sound absorption and transmittance. All polygon shapes and positions are described in terms of a 3D co-ordinate system or 'scene graph'. To view this scene graph the objects must be transformed relative to the observer's position and viewing direction via matrix transformations. This 3D model view is then shown on a 2D window via perspective and view port transformations. To hear the effects of these polygons they are similarly transformed relative to the listener's position and direction. For 'first person' applications, observer and listener positions and directions are the same.

In constructing a virtual environment, the function calls for visual and acoustic components both require initialisation functions followed by transformations each time the user moves. For instance, the following OpenGL code sets up viewing parameters for the observer...

```
Initialise_Graphics()
{
        //initialise graphics
        ....
        // Set Up Observer
        glViewport(0, 0, 640, 480);
        glEnable(GL_DEPTH_TEST);
        glMatrixMode (GL_PROJECTION);
        glLoadIdentity ();
        gluPerspective(ViewAngle,1,.1,300);
        glMatrixMode (GL_MODELVIEW);
        glLoadIdentity();
        ....
}
```

The following A3D code initialises listener and source positions.

```
Initialise_Audio()
{
        //Initialise acoustics
        ...
        //Set Sources
        ...
        a3droot->NewSource(A3DSOURCE..., &a3dsrc[i]);
        a3dsrc[i]->LoadWaveFile(filename);
        a3dsrc[i]->SetPosition3f(x,y,z);
        a3dsrc[i]->Play(A3D_LOOPED);
        ....
        // Set Up Listener
        a3dgeom->LoadIdentity();
        a3dlis->SetPosition3f(X,Y,Z);
```

```
        a3dlis->SetOrientationAngles3f(THETA, 0.0f, 0.0f);
}
```

When the user moves around in the 3D graphics world polygons will be rotated and transformed relative to the current observer position and orientation. Polygons are specified by their vertices and visible material properties as given by textures. For instance, the following OpenGL code specifies a visible polygon.

```
Draw_Polygon(X,Y,Z,THETA)
{
        glPushMatrix();
        glRotatef(-THETA,0,1,0);
        glTranslatef(-X,-Y,-Z);
        glTexImage2D(GL_TEXTURE_2D, ...texture image data);
        glTexParameterf(GL_TEXTURE_2D,...how the texture is wrapped);
        ....
        glBegin(GL_POLYGON);
          glVertex3f(0,0,0);
          glVertex3f(10,0,0);
          glVertex3f(10,10,0);
          glVertex3f(0,10,0);
        glEnd();
        glPopMatrix();
        ....
        glFlush();
}
```

Similarly, as the user moves around the acoustic world polygons are transformed relative to the current listener position. These polygons being specified by their vertices and acoustic material properties like absorption and transmittance. The following A3D code specifies an acoustic polygon.

```
Draw_Acoustic_Polygon(X,Y,Z,THETA)
{
        a3dgeom->PushMatrix();
        a3dgeom->Rotate3f(-THETA,0,1,0);
        a3dgeom->Translate3f(-X,-Y,-Z);
        a3dgeom->BindMaterial(...acoustic texture data);
        ....
        a3dgeom->Begin(A3D_QUADS);
          a3dgeom->Vertex3f(0,0,0);
          a3dgeom->Vertex3f(10,0,0);
          a3dgeom->Vertex3f(10,10,0);
          a3dgeom->Vertex3f(0,10,0);
        a3dgeom->End();
        a3dgeom->PopMatrix();
        ....
        a3droot->Flush();
}
```

As well as basic polygon rendering both APIs provide optimisation through display lists where the results of functions can be collected and repeated later. Also collections of planes can be constructed relative to their own co-ordinated systems and easily transformed as objects into the main scene graph so that complex, dynamic environments can be created. The similarity of OpenGL and A3D allows for fast and easy integration of graphics and audio in the application code.

2.4 Implementation for Teaching Acoustics

Already there are numerous applications available that combine OpenGL and Aureal A3D notably recent P.C. based simulations and games where localisation helps the user respond to events and threats while the prediction of room acoustics adds to mood and ambience. Similar techniques can be used to develop worth while Design and Teaching applications. It is the aim of the author to develop interactive virtual environments for the teaching of acoustics. The user will be able to design and modify rooms and position sound sources, objects, reflectors, barriers, etc whilst gaining instant acoustic feed back. This could give the student a more intuitive feel for room acoustics whilst by providing useful information on how to design and layout public and working spaces for optimal comfort and worker efficiency.
Figure 2 shows the modelling of Salford University's Dept. of Acoustics staff rooms. The user can move from room to room whilst appreciating how various sources (as shown by diamonds) sound relative to the reflections and occlusions of walls and furniture. The application lets the user

- import and amend architectural data (i.e. surface dimensions and material data)
- insert, locate and orientate furniture objects selected from a wide range of different types (e.g. chairs, doors, bookcases, cupboards, etc)
- specify and position of multiple sound sources selected from a wide range of different types (male and female voices, radios, kettles, photocopiers, etc)

2.5 Evaluation of Teaching Software

Clearly, the application's worth will be based on effectiveness in simulating and teaching room acoustics. The following may be used for assessment

- comparison of predicted sounds with and real recorded sounds by modelling real rooms within the application
- implementation of other non-real time prediction methods for further comparison
- subjective impressions from users to assess accuracy
- user surveys to asses the application's ease of use and educational value

3. THE FUTURE

Even with hardware acceleration, A3D real-time acoustic prediction for most applications is limited by processor bandwidth to first order reflections with a reverberation tail. However, there is support for higher orders of reflection and as chip manufacture switches to 0.18 and later 0.13 micron etching technologies the quality of acoustic prediction will improve.
Currently the A3D allows absorption and transmission coefficients to be specified by high and low frequency components, though future implementations will take onboard multiple frequency bandwidth absorption and transmission coefficients. There is also scope for diffusion and diffraction modelling though all such developments are driven by the requirements and requests of end users and the acoustic community.

It is worth mentioning Creative Labs Environmental Audio Extensions (EAX 2.0) API for their very popular 'Soundblaster Live' sound cards. This API is considered the main competitor to A3D being also capable of localisation and occlusion detection. However true room acoustics prediction has been sacrificed for a simpler to program reverberent effects simulator.

Future graphical and acoustic API's will aim to provide higher levels of functionality whilst optimising for speed and portability. For instance, the forthcoming Fahrenheit API is a joint development by the currently competing Silicon Graphics and Microsoft corporations. Fahrenheit will allow the developer to manipulate and render to the scene graph directly without worrying about more esoteric issues such as transformations and special optimisations. The API intends to make the integration of audio easy providing a high level wrapper to DirectSound and possibly A3D. All this will allow the developer to concentrate less on programming and more on content.

4. REFERENCES

1 Segal & Akeley, "The Design of the OpenGL Graphics Interface", Silicon Graphics Incorporated, http://www.opengl.org/

2 Microsoft, "About DirectX", http://www.microsoft.com/directx/

3 Aureal, "A3D White Paper", Aureal Semiconductor, www.aureal.com and www.a3d.com

4 A. Kulowski, "Algorithmic Representation of the Ray Tracing Technique," Applied Acoustics 18, 449-469, (1984)

5 H. Lehnert, "Systematic Errors of the Ray-Tracing Algorithm," Applied Acoustics 38, 207-221, (1993)

6 J. Borish, "Extension of the Image Model to Arbitrary Polyhedra," J. Acoust. Soc. Am. 75 (6), 1827-1836, (1984)

7 H. Lee & B. Lee, "An Efficient Algorithm for Image Method Technology," Applied Acoustics 24, 87-115, (1988)

8 M. Volander, "Simulation of Transient and Steady State Sound Propagation in Rooms using a New Combined Ray Tracing / Image Source Algorithm," J. Acoust. Soc. Am., 86 (1), (1989)

9 D. Van Maercke & J. Martin, "The Prediction of Echograms and Impulse Responses within the Epiduare Software", Applied Acoustics 38, 93-114, (1993)

10 T. Lewers, "A Combined Beam Tracing and Radiant Exchange Computer Model of Room Acoustics," Applied Acoustics," 38, 161–178, (1993)

11 I.Drumm, "The Development and Application of an Adaptive Beam Tracing Algorithm to Predict the Acoustics of Auditoria", PhD Thesis, The University of Salford, UK, 1998

12 Hogdson, Heerema, Halingten
PlantNoise sysem for the prediction, visulisation and auralization of industrial noise during computer walk-through
JASA Feburary 1999 - Volume 105, Issue 2 p. 989

13 Funkhouser
Feb 1999
Interactive acoustic modelling in complex environments
JASA Feburary 1999 - Volume 105, Issue 2 pp. 1357-1358

14 Kutruff
Auralisation of Impulse Responses Modeled on the Basis of Ray-Tracing Results
J. Audio Eng Vol 41, No 11, 1993 November

15 Rainer Feistel
Ears Auralisation Software
J. Audio Eng Vol 41, No 11, 1993 November

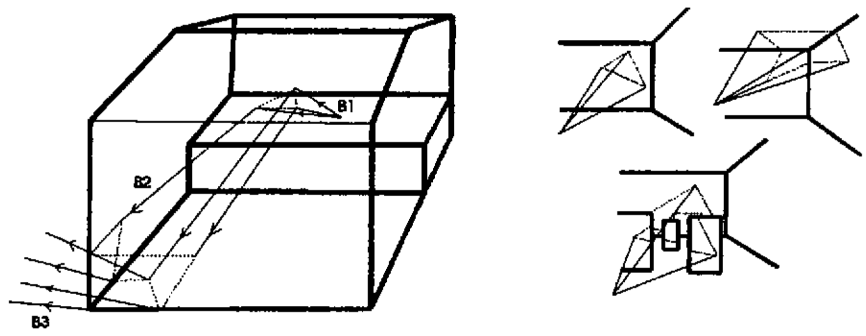

Fig.1. The adaptive beam tracing algorithm models sound as an omni-directional spread of beams whose reflected child beams have their cross-sectional shapes determined by areas of illumination. The first beam (B1) is reflected to create a child beam (B2), B2 in turn spawns numerous child beams such as B3 formed from the sections of planes illuminated. The algorithm handles occlusion and awkward scenarios.

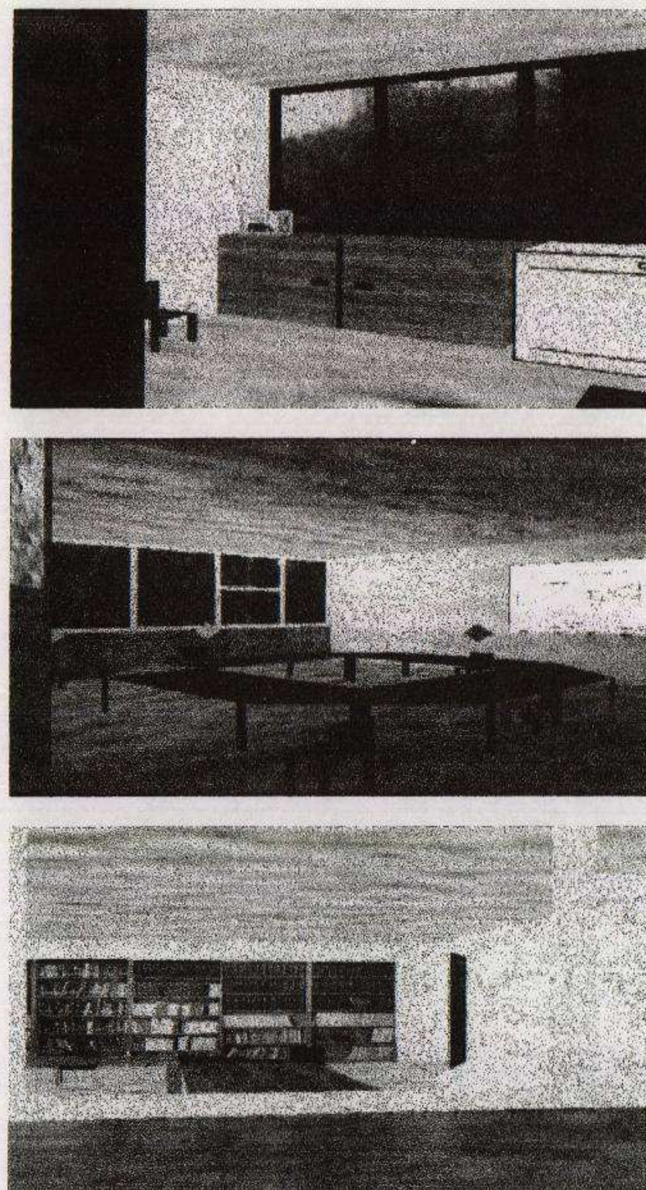

Fig.2. The modelling of Salford University's Department of Acoustics staff rooms with OpenGL and A3D. The diamonds shown represent sound sources the user hears relative to position and room acoustics.