Binaural Simulations Using Audio Rate FDTD Schemes and CUDA

BINAURAL SIMULATIONS USING AUDIO RATE FDTD SCHEMES AND CUDA

*Craig J. Webb**
Acoustics Group/Edinburgh Parallel Computing Centre
University of Edinburgh, UK
C.J.Webb-2@sms.ed.ac.uk

Stefan Bilbao
Acoustics Group
University of Edinburgh, UK
sbilbao@staffmail.ed.ac.uk

ABSTRACT

Three dimensional finite difference time domain schemes can be used as an approach to spatial audio simulation. By embedding a model of the human head in a 3D computational space, such simulations can emulate binaural sound localisation. This approach normally relies on using high sample rates to give finely detailed models, and is computationally intensive.

This paper examines the use of head models within audio rate FDTD schemes, ranging from 176.4 down to 44.1 kHz. Using GPU computing with Nvidia's CUDA architecture, simulations can be accelerated many times over a serial computation in C. This allows efficient, dynamic simulations to be produced where sounds can be moved around during the runtime. Sound examples have been generated by placing a personalised head model inside an anechoic cube. At the lowest sample rate, 44.1 kHz, localisation is clear in the horizontal plane but much less so in the other dimensions. At 176.4, there is far greater three dimensional depth, with perceptible front to back, and some vertical movement.

1. INTRODUCTION

In *virtual acoustics* applications, direct numerical methods are increasingly becoming a viable approach. Full wave-based techniques can give highly detailed and accurate results, and the finite difference time domain method (FDTD) is particularly suitable for acceleration using GPU computing [1]. Such an approach allows the possibility of 3D localization, giving binaural results when using stereo headphones.

Research into the simulation of head-related transfer functions (HRTFs) has been conducted using techniques such as boundary element methods [2], and indeed finite difference methods [3]. This research has focused on calculating accurate HRTFs, based on generic or individual scans of the human head. In the case of FDTD, very high sample rates are used (up to 500 kHz) in order to discretize the models at a fine level of detail [4].

This paper investigates 3D localization within *audio rate* FDTD simulations, from 44.1 to 176.4 kHz. Rather than computing static HRTFs based on impulse responses, sound can be localized within a fully dynamic virtual acoustic environment. These simulations allow the movement of sound sources during runtime, which is difficult to achieve using impulse response rendering, requiring interpolation.

Section 2 outlines the discretization of head geometry over a uniform grid. Section 3 describes the FDTD scheme employed, and Section 4 its implementation using CUDA. Results are presented in Section 5.

* This work was supported by the European Research Council, under Grant StG-2011-279068-NESS

2. 3D HEAD MODEL

The first stage in creating a binaural FDTD simulation is to obtain an accurate model of the human head. In order to minimise the inter-personal effect of HRTFs in testing, a model was constructed from scans of the primary author's own head. This was produced using a stereo-photographic process, giving 3D scans from various angles which are then combined to give a complete model[1]. The result is a dense point cloud of coordinates indicating the surface of the head.

With a view to parallelization in CUDA, the point cloud was voxelized into a 3D data grid. The grid density is set according to the stability condition for the FDTD scheme (see Section 3) which dictates a minimum spacing X for a given audio sample rate F_s. (In this article, a simple scheme is employed, so the relationship to be satisfied is $X \geq \sqrt{3}c/F_s$, where c is the wave speed in air.) The data was voxelized using the following algorithm:

1. Iterate through the point cloud list, scaling the coordinates to the required grid spacing and rounding them to the nearest point in the 3D data grid.
2. Identify all points on or outside the surface boundary, by tracing rays from each face of the minimum parallelepiped surrounding the head. What is left is considered to be the interior of the head.
3. Calculate the correct FDTD scheme coefficients for all points not inside the surface.

The voxelization for a grid spacing of 3.4mm (176.4 kHz) shows detailed resolution (Figure 1). Facial features and pinna definition are readily apparent.

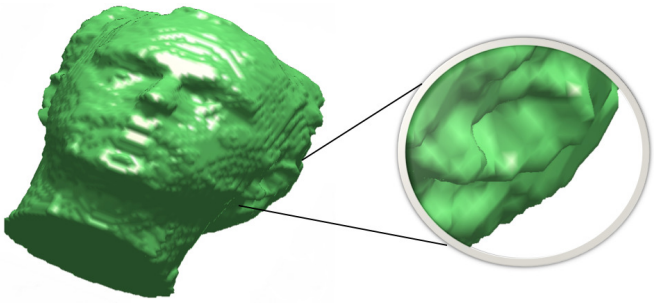

Figure 1: *Voxelization at 3.4mm for 176.4 kHz, with a magnified view of the left pinna.*

[1] Stereo-photographic scanning provided by Prof. R. Fisher and S. McDonagh, School of Informatics, University of Edinburgh.

A sample rate of 44.1 kHz requires a grid spacing of at least 13.5mm, Figure 2 shows the result. Whilst the basic shape of the head is still clear, the detail of the pinna is almost completely lost. At 88.2 kHz, the grid spacing of 6.8mm gives some definition of the shape of the pinna, certainly better than at 13.5mm. The effect of the different voxelizations was tested by embedding the head models into a 3D space, using a single variable FDTD scheme. This incorporates basic reflective boundaries on the head surface with loss. The rectangular cuboid containing the head data is placed inside a larger space with numerically absorbing boundary conditions, as shown in Figure 3.

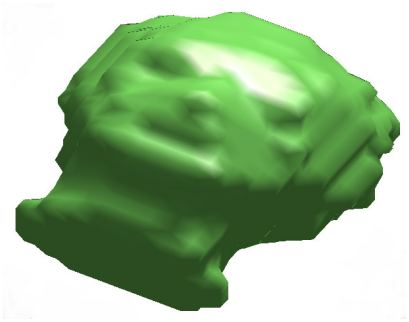

Figure 2: *Voxelization at 13.5mm for 44.1kHz.*

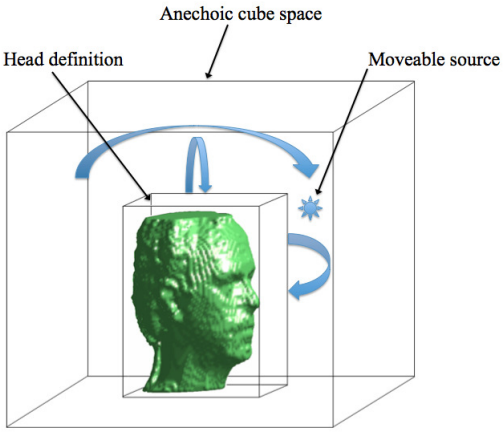

Figure 3: *Simulation setup in anechoic FDTD cube.*

Sound sources were input into the scheme during runtime, and then dynamically moved around inside the simulation space.

3. FINITE DIFFERENCE SCHEME

Acoustic propagation is modelled using the second order 3D wave equation:

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

Here, $u(x, y, z, t)$ is the target acoustical field quantity, c is the wave speed in air, and ∇^2 is the 3D Laplacian. The simplest possible FDTD scheme is of the "seven point" variety []:

$$u^{n+1}_{l,m,p} = (2 - 6\lambda^2)u^n_{l,m,p} + \lambda^2 S^n_{l,m,p} - u^{n-1}_{l,m,p} \tag{2}$$

where $\lambda = \frac{cT}{X}$ ($T = F_s$ is the time step and X the grid spacing), and $S^n_{l,m,p}$ is the sum of the six direct neighbouring points in u^n. The stability condition is derived using von Neumann analysis [5], leading to the constraint mentioned above:

$$X \geq cT\sqrt{3} \tag{3}$$

Two types of boundary condition were used. First, the surface of the head is assumed to be rigid and a lossy condition is used:

$$\frac{\partial u}{\partial t} = c\beta \mathbf{n} \cdot \nabla \mathrm{u} \tag{4}$$

where $\mathbf{n}$ is the normal to the surface at the head boundary, and β is a simple loss coefficient. The integer 6 from the interior update is reduced for each of the neighbouring points of u^n which fall inside the boundary, and $S^n_{l,m,p}$ adjusted accordingly. It is this integer value which is stored in the head definition data.

To create an anechoic boundary around the cube, an absorbing condition is required. Whilst a perfectly matched layer is the standard approach from electro-magnetics [6], a computationally cheaper method can be employed based on the Enquist Majda condition [7]. If $\mathbf{n}$ is a normal to a planar boundary, then the condition can be written as:

$$\left(\frac{\partial}{\partial t} + c\mathbf{n} \cdot \nabla\right)^q u = 0 \tag{5}$$

where q determines the order of the approximation. Although this gives some minor reflections for high frequency waves which are tangential to the boundary, informal listening tests suggest that the conditions are reasonably transparent at audio sample rates, even with an order as low as $q = 2$.

For a dynamic source location, appropriately sampled audio signals were summed into the grid at each time step using tri-linear interpolation. The input position was calculated along a 3D path defined by numerous points inside the space. Tri-linear interpolation provides a simple, but effective, method for moving the input, with minimal perceptible artefacts.

4. IMPLEMENTATION USING CUDA

Initial prototyping was performed in MATLAB. However, even at the lowest sample rate of 44.1kHz and for a cube of less than $1\mathrm{m}^3$, a serial C code takes around a minute for one second of simulation. At 176.4 kHz, the computation is 256 times larger and takes hours. Using GPU computing and Nvidia's CUDA architecture, one can greatly reduce the runtime [8]. Porting the code is simplified using the latest CUDA version 4.1 and the FERMI GPU cards.

CUDA version 4 enables a three dimensional grid for the thread blocks [9], allowing a direct mapping of the 3D data to block indices. Using a FERMI architecture card, complex shared memory coding is unnecessary as the caching system provides the same function [10]. Each thread can read data directly from global memory, and setting the device to prefer the L1 cache gives a further improvement in performance.

Achieving optimal performance still relies on global memory *coalescing*. This is a particular issue for the exterior boundary updates of the cube, where the faces and edges necessarily have different orientations and their data is not always contiguous in memory. Considerable speedup was obtained by rearranging the left and right faces into temporary arrays to facilitate coalescing (Figure 4). Each Z-layer of these non-contiguous faces was loaded

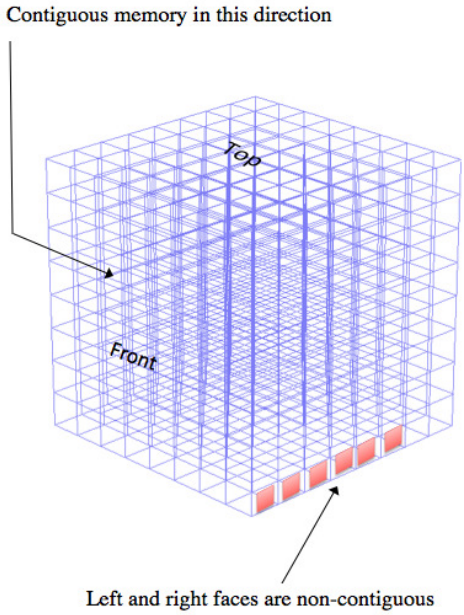

Figure 4: *Non-contiguous memory on left and right faces.*

into linear memory, which can then be accessed in a contiguous manner when applying the update to the face boundaries. This rearranging of data can also be masked by performing it during the data loading for the interior points.

For the interior and head data updates, a two stage approach was used. An initial kernel performs an update over the entire interior space, followed by a second kernel which corrects the grid for the head definition. A further optimisation was possible for the 44.1 and 88.2 kHz simulations, as the head definition data could be stored as single byte *char*s in constant memory.

5. RESULTS

Testing was performed on a single Nvidia Tesla C2050 card at double precision, for an outer cube of width 86 cm. Table 1 shows the resulting computation times for one second of simulation. Moving

Sample Rate (kHz)	Total Grid points	Time (sec)
44.1	262,144	6.0
88.2	2,097,152	70.3
176.4	16,777,216	1,360.6

Table 1: Computation times for 1 second of simulation.

from 88.2 to 176.4 kHz takes nearly twenty times longer, as the head definition data is read from global memory.

Initial audio tests used dry recordings of a matchbox and also a short vocal sample. During a six second simulation, this audio signal was moved around the head on a flat circular path and also an angled path as shown in Figure 5. The left and right readout positions were taken from individual grid points on the boundary of the head, approximately at the entrance to the ear canal. Note that the ear canal itself was not modelled due to the coarse nature

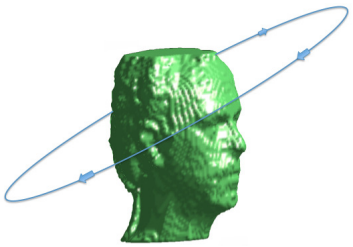

Figure 5: *Source movement used for initial testing.*

of the grid used. Audio files are available at `http://www2.ph.ed.ac.uk/~s0956654/Site/VirtualRoomAcoustics.html`

Starting at 44.1 kHz, movement around the azimuth angle of the horizontal plane is immediately apparent and the position can be judged. What is less clear is the location either in front or behind the head, and variation in elevation angle is only slightly discernible at the front. At 88.2 kHz, the result is similar to the previous ones, but azimuth variation is more clearly defined. At 176.4 kHz there is a distinct change and spatialisation effects are more easily discernible. Azimuth variation is very clear, and there is sense of the movement from front to back, along with some effects of changes in elevation. At all sample rates, the head definition gives spatial effects which are clearly absent from the simulations using just an empty space.

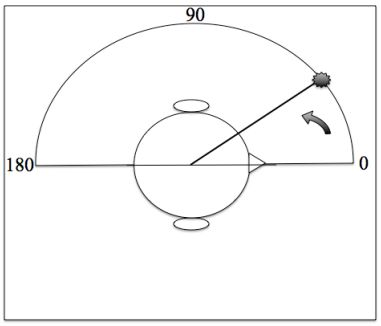

Figure 6: *Semi-circular path at 40cm from centre of head.*

Analysing the system using impulse response simulations can be performed to derive the head-related impulse response (HRIR) and HRTF for various source positions. Initial tests were performed using impulses at five degree increments around a semi-circlular path (Figure 6). Results are shown in Figures 7 and 8.

6. CONCLUSIONS

3D audio localization is of course based on various features: objective factors such as the type of sound source used, and also necessarily subjective factors such as the individual nature of the HRTF. The purpose of this initial study was to examine if any form of localization was perceivable using audio rate FDTD schemes and basic boundary conditions.

At the lower sample rates of 44.1 and 88.2 kHz, only the variation in the horizontal plane is clearly defined. Movement in elevation and judging back to front is difficult. The audio results at 176.4 kHz are much improved, giving clear localisation in the

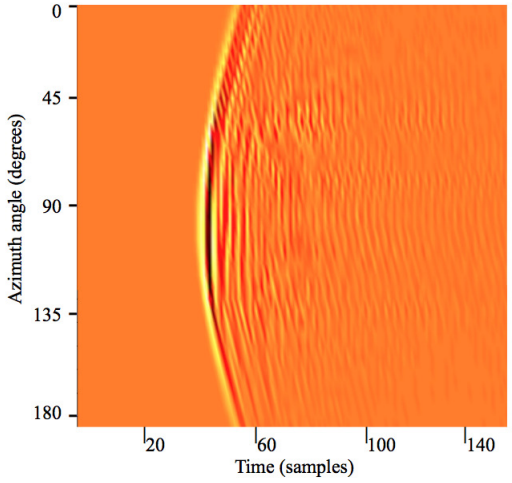

Figure 7: *HRIR for left ear at 176 kHz.*

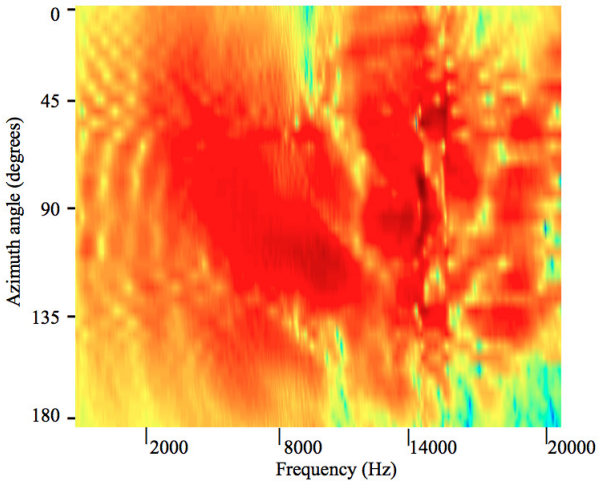

Figure 8: *HRTF for left ear at 176 kHz.*

horizontal plane as well as perceptible back to front movement. Changes in elevation are apparent but harder to accurately place, and would benefit from the inclusion of the upper torso in the model.

Running simulations at sample rates greater than 44.1 kHz is far less efficient, increasing the workload by x16 for each doubling of the rate. For large scale virtual acoustics modelling, this is clearly not advantageous, both in terms of computation times and indeed memory usage. Further development using more refined boundary conditions for defining and voxelizing the head model should allow the effects of the 176.4 kHz simulation to be heard at the lower sample rates. An open question is the level of detail necessary in the representation of the pinna. Future work will be concerned with perceptual testing, with regard to the issues mentioned above.

7. REFERENCES

[1] L. Savioja, D. Manocha, and M. Lin, "Use of GPUs in room acoustic modeling and auralization," in *Proc. Int. Symposium on Room Acoustics*, Melbourne, Australia, Aug 2010.

[2] Y. Kahana and P. Nelson, "Boundary element simulations of the transfer functions of human heads and baffled pinnae using accurate geometric hearing," *J. Sound and Vibration*, vol. 300, pp. 552–579, 2007.

[3] P. Mokhtari, H. Takemoto, R. Nishimura, and H. Kato, "Computer simulation of HRTFs for personalization of 3D audio," in *Proc. International Symposium on Universal Communications*, Osaka, Japan, Dec 2008.

[4] M. Nakazawa and A. Nishikata, "Develompent of a sound localization system with tube earphone using human head model with ear canal," *IEICE Trans. Fundamentals*, vol. E88-A(12), pp. 3584–3592, Dec 2005.

[5] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks/Cole Advanced Books and Software, Pacific Grove, California, 1989.

[6] J.P. Berenger, "Three-dimensional perfectly matched layer for the absorption of electromagnetic waves," *J. Comp. Phys*, vol. 127(2), pp. 363–379, Sept 1996.

[7] B. Enquist and A. Majda, "Absorbing boundary conditions for the numerical evaluation of waves," *Mathematics of Computation*, vol. 31(139), pp. 629–651, 1997.

[8] C.J. Webb and S. Bilbao, "Computing room acoustics with CUDA - 3D FDTD schemes with boundary losses and viscosity," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, May 2011.

[9] Nvidia Corp, "CUDA C Programming Guide v4," Available : http://developer.nvidia.com/cuda-toolkit-41, May, 2011.

[10] Nvidia Corp, "Fermi compute arch. whitepaper," Available : http://www.nvidia.com/object/fermiarchitecture.html, Oct, 2010.