

# The Matrix: Integrating Audio, Visual and 3D Structure in a Real-Time Virtual World

Bart Thomee    Michael S. Lew  
LIACS Media Lab  
Leiden University  
{bthomee, mlew}@liacs.nl



**Keywords:** 3D audio, virtual reality, virtual environments, networked multipresence

## ABSTRACT

In the current work on multiplayer virtual reality, the research has focused largely on the visual aspect combined with text. Audio is often neglected and when present typically ignores the structure of the virtual world. In this work, we present a new system which integrates the audio, visual, and 3D structure of the virtual world. Specifically, our novel contribution is creating a system which models the effect of the 3D world structure upon the audio and visual aspects in a natural and intuitive manner for a massive multiplayer world. For example, when a group of people are talking in the next room, the sound will not be heard due to the structure of the wall. By having the audio and video affected by the structure of the world, we are able to create a better immersive multiplayer virtual experience.

## 1 INTRODUCTION

Visions of the future such as The Matrix [1], The Street (from the book Snow Crash [2]), or Cyberspace (from the book Neuromancer [3]) all support audio and visual communication in a way which works naturally with reality. In the virtual world, when someone walks around a corner, you no longer see him nor hear him.

The promise of virtual reality has been within the public consciousness for decades, however, the technology for achieving an immersive experience has only been available fairly recently. Multimedia hardware, such as 3D graphics cards, real time audio codecs, and moderate upload and download bandwidth are necessary. Currently, resources such as client bandwidth are limited. Furthermore, many users do not have static IP addresses or are located behind routers which do not allow direct externally initiated connections. Fundamentally, we wanted to allow the most number of people over the world to connect to our system, thus we decided to focus on a system which would satisfy the following constraints.

- Allow a large number of clients (>100) from the Internet
- Require only moderate client bandwidth (~128kbps)
- Not require a static IP address
- Work across a bi-directional NAT

In this paper, we present a virtual reality framework that allows players in virtual environments to talk with each other as in real life. As its naturalness in communication resembles that of the future world as sketched in the movie The Matrix, we hence entitled our framework with the same name. By embedding our framework into a 3D

engine, it has access to the positioning of all clients and also to the physical structure of the world. As a result, the framework enables audio to be correctly attenuated using the distance and angle between clients that are talking, while also taking interference from environmental objects into account. We have devised a novel audio attenuation algorithm for partial structural occlusion based on Cauchy’s probability distribution in order to estimate the amount of volume reduction while navigating within the environment.

The current research literature as seen from the ACM and IEEE digital libraries appears to indicate that there have been no published systems which integrated audio, video, and 3D structure with a massive multi-client environment (i.e. >100 clients) in a realistic manner [4-9]. For example, in the work by Boustead, et al. [4], they also use the Quake3 engine but do not model the interference of the 3D walls in the virtual world or levels.

The structural audio problem occurs when the 3D structure interacts with the audio signal. Examples include simply going around a corner or walking into a room and closing the door. In both cases, the 3D structure affects the audio – typically lowering the amplitude but potentially also causing audio reflections and refractions. In some ways it is more complex than the hidden surfaces problem in that in most cases hidden surfaces are simply not drawn by the renderer. For structural audio, we can not simply cut off the audio. We must have a natural drop off due to the interference with the 3D world.

## 2 AUDIO AND VISUAL FRAMEWORK

### 2.1 Architecture

Several architectures exist for the exchange of audio between multiple clients, for example peer-to-peer, centralized server and distributed server [9]. To fulfill the constraints mentioned in Section 1 and to keep the system as simple as possible, we opted for the centralized server architecture. In order to maximize portability, the audio framework is implemented as a DLL-based plugin and therefore can be embedded into any virtual world. This does mean however that the existing in-game audio processing capabilities are not used.

An interesting method has been proposed in [10] to adapt to variable amount of speakers and network congestion by having the server dynamically pre-

mix one or more audio streams for individual clients. The downside of this approach is that not only the burden on the server increases, but more importantly that clients won’t be able to separate these mixes into distinct voices and correctly position them in 3D space. Thus in our framework, mixing of voices takes place at the client-side to ensure that each client can create its own spatial audio mix, despite requiring a higher bandwidth usage. Depending on their bandwidth limitations and how severe the network is congested, clients notify the server to reduce or increase the amount of audio streams. Lost packets are recreated using the insertion-based repair technique involving repetition and fading [11], and soundcard clock skew [12] is compensated for. To ensure high sound quality while maintaining a reasonable bandwidth footprint, voices are encoded using the Speex [13] codec in wideband mode.

### 2.2 Positional and Structural Audio

In previous research, sophisticated methods [7] have been described for real-time audio modeling in distributed virtual environments. However, the previous methods typically had the assumption of non-moving audio sources and required a lot of processing power. In contrast, our intention is to examine the problem where all of the audio sources are moving in real-time and modeling the audio only requires low computational complexity.

The main feature of the framework is a novel audio algorithm that deals with handling partial structural occlusion. To determine this attenuation factor that should be applied to a sound– in addition to distance and angle attenuation – the algorithm employs Cauchy’s probability distribution to weight a grid that is placed with its center at the listener’s location, pointing at the origin of the sound. Cauchy’s probability density function, with the peak of the distribution at  $x = 0$ , is specified as

$$f(x, \gamma) = \frac{1}{\pi} \left( \frac{\gamma}{x^2 + \gamma^2} \right) \quad (1)$$

where  $\gamma$  is the scale parameter ( $\gamma > 0$ ), which specifies the half-width at half-maximum.

Our weight function takes three steps to determine its grid values:

- i. Assume the grid is oriented along the  $v$  and  $w$  axes, with the origin at the center point of the grid. Convert the location of the grid point in question to radius  $r$

$$r(v, w) = \begin{cases} |v| & \text{if } |v| \geq |w| \\ |w| & \text{otherwise} \end{cases} \quad (2)$$

- ii. Determine the Cauchy grid value. This is done by entering the radius obtained by (2) into (1). For any radius other than zero the result is first doubled due to the symmetrical nature of the distribution and then averaged over the amount of grid points belonging to the specified radius. This is expressed by

$$c(r, \gamma) = \begin{cases} \frac{1}{\pi\gamma} & \text{if } r = 0 \\ \frac{1}{4\pi r} \left( \frac{\gamma}{r^2 + \gamma^2} \right) & \text{otherwise} \end{cases} \quad (3)$$

- iii. The current grid values will not sum up to one, as the tail sections of the Cauchy distribution have not been considered yet. Therefore, every grid value should be increased by a small quantity  $\delta$ , where

$$\delta = \frac{1 - \sum_{v=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} \sum_{w=-\lfloor N/2 \rfloor}^{\lfloor N/2 \rfloor} c(r(v, w), \gamma)}{N^2} \quad (4)$$

This results in the weight function

$$g(v, w, \gamma) = c(r(v, w), \gamma) + \delta \quad (5)$$

The ‘audibility’ of each point on the grid is determined by tracing the visibility between itself and the sound, see Figure 1. The attenuation factor is formed by adding only the weights of the grid points that are ‘audible’. This technique results in smooth sound transitions when moving around objects and corners while talking to other players.

Our audio algorithm utilizes the Cauchy distribution because it (a) has been shown in other areas to be more realistic to real world distributions [14] and (b) in the future will allow us to adaptively adjust its parameters, e.g. simulate different environments or modifying sound perception through the use of in-game items.

With our audio framework, players can have conversations with many people at the same time, as the audio correctly appears to originate from the visual location of the players that are talking.

Moreover, players are able to localize any sound source and direct visual attention to where the sound is coming from. Note that our method does not take reflections, refractions and interference with other sound waves into account.

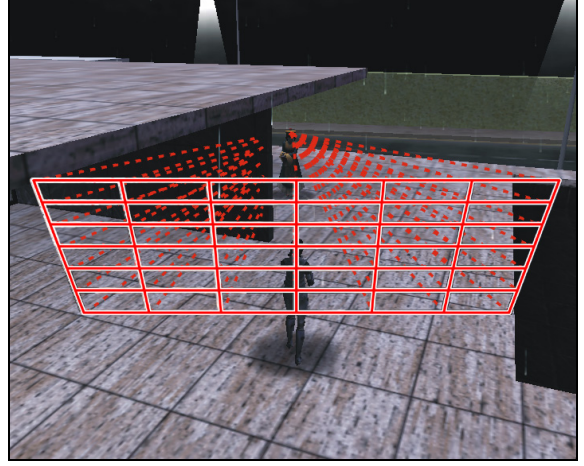


Figure 1. Tracing audibility

### 3 EXPERIMENTS

The experiments described in this section were performed by revising the Quake III: Arena source code to correctly handle thousands of clients and adding the DLL-based plugin for the structural audio framework. We chose the Quake engine because the clients required the least network traffic; there are open source 3D editing tools for the 3D worlds available; and it natively has the option to split the video signal into binocular components for a 3D experience using either shutter glasses or a head mounted display.

#### 3.1 Data Collection

The client systems involved in the experiments had Pentium 4 processors in the 1,8-3,0 GHz range with 512-1024 MB RAM and contained generic video and sound cards. The client network connections were a mixture of LAN and ADSL connected to a 3.4 GHz P4 server.

In a time span of approximately six hours, 100 clients joined the server. Initially, the groups were kept small, but with increasing amounts of clients, groups grew larger. Note that this also indicates that ‘more connected clients’ corresponds with ‘more nearby clients’ and vice versa. This reasoning is used when illustrating the experimental results.

Latency could be determined only with a resolution of 10ms.

The selected 3D environment was a reconstruction from plans of Mies van der Rohe's German pavilion for the 1923 Barcelona Exhibition.

### 3.2 Evaluation

Our two primary goals were to (i) measure quantitatively the quality of the audio over the network with a focus on increasing numbers of clients and (ii) qualitatively ask the human users if the audio experience was immersive.

In Figure 2 we have plotted the measured latency against the number of clients that a client had within hearing range. ITU-T Recommendation G.114 [15] suggests that the mouth-to-ear delay, or latency, should not exceed 400ms and preferably be lower than 150ms. We can conclude that even though the latency increases with a growing number of clients, it remains within acceptable limits. In comparisons with typical mobile phones between students, the latency was found to be subjectively lower and the audio quality better. The curve of the graph might indicate that in overly populated areas the latency can grow unacceptably large; in the future we will perform experiments with more clients.

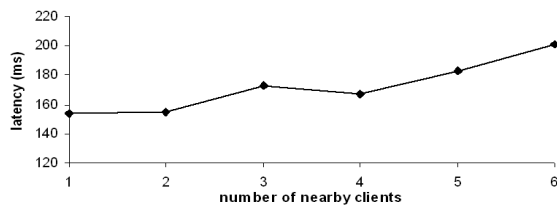


Figure 2. latency vs. number of nearby clients (client-side)

Figure 3 shows the packet loss against the number of nearby clients. Generally, the acceptability of an audio signal experiencing packet loss depends on the codec with which the signal is encoded, but the figure indicates that the amount of packet loss is small enough to have little to no effect on the audio quality. Packet loss at the server was negligible.

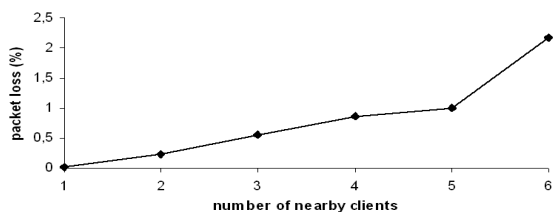


Figure 3. packet loss vs. number of nearby clients (client-side)

In a separate smaller scale session, we have also tested with intercontinental ADSL connections (typically 1.5Mbps downlink, 500kbps uplink) and found that the packet loss increases based on the distance. For the clients connecting from the USA to the Netherlands, there was approximately 1.3 percent packet loss.

The total amount of CPU usage by the server, i.e. by the Quake server with the audio framework embedded, is graphed in Figure 4 against the number of connected clients. We notice that the CPU usage increases linearly with a growing amount of clients. Only a slight increase in memory usage by the server can be noticed in Figure 5, which is due to the allocation of new client objects in the audio server. Both these figures suggest that many more clients can be supported easily, provided a more powerful server to handle the expected increase in CPU usage.

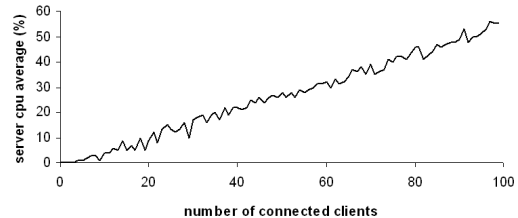


Figure 4. CPU usage vs. number of connected clients (server-side)

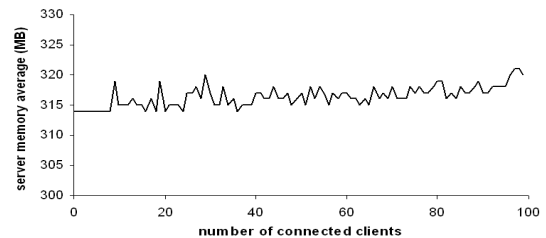


Figure 5. memory usage vs. number of connected clients (server-side)

Overall, the users were pleased with the performance of the system and felt that our communication framework delivered an extra layer of immersion in the virtual world.

## 4 DISCUSSION AND CONCLUSIONS

In this paper, we presented a virtual reality environment which features a novel structural audio algorithm providing immersive sound attenuation through its knowledge of the structure of the virtual world. The underlying Cauchy-based audio framework ensures smooth sound transitions when moving around objects and corners and only requires

minimal CPU effort for real-time modeling. The key strength of our audio algorithm lies in its efficiency, as our user experiments have shown that convincing audio attenuation can be attained with low computational complexity.

In the future we will be shifting to a quad core system and large scale internet testing. One important usability feature will be a waiting queue for when there are more clients than the server can handle in real time. While we have not tested aspects such as server stability over periods of months, the current version is subjectively moderately stable and it is our intention to demonstrate the system to the conference attendees. In the future stage of experiments we will also explore the interest level of different usage contexts: international meeting place, help desk, virtual conferences, virtual classrooms and perform experiments to determine the effectiveness of our system.

## REFERENCES

- [1] A. Wachowski, L. Wachowski, "The Matrix", Warner Home Video, 1999
- [2] N. Stephenson, "Snow Crash", Bantam Spectra, 1992
- [3] W. Gibson, "Neuromancer", Ace Books, 1984
- [4] P. Boustead, F. Safaei, M. Dowlatshahi, "DICE: internet delivery of immersive voice communication for crowded virtual spaces", in *IEEE Virtual Reality Conference 2005*, pp. 35-41, 2005.
- [5] C. Greenhalgh, S. Benford, "MASSIVE: A collaborative virtual environment for teleconferencing", in *ACM Transactions on Computer-Human Interaction*, pp. 239-26, 1995.
- [6] J. Robinson, J. Stewart, I. Labbe, "MVIP - audio enabled multicast VNet", in *Proc. of Symposium on Virtual Reality Modeling Language*, pp. 103-109, 2000.
- [7] T. Funkhouser, P. Min, I. Carlbom, "Real-time acoustic modeling for distributed virtual environments", in *Proc. of the 26th annual conference on Computer Graphics and Interactive Techniques*, pp. 365-374, 1999.
- [8] M. Naef, O. Staadt, M. Gross, "Spatialized audio rendering for immersive virtual environments", in *Proc. of the ACM Symposium on Virtual Reality Software and Technology*, pp. 65-72, 2002.
- [9] C. Nguyen, F. Safaei, D. Platt, "On the provision of immersive audio communication to massively multi-player online games", in *Proc. of the IEEE ISCC 2004*, pp. 1000-1005, 2004.
- [10] M. Radenkovic, C. Greenhalgh, S. Benford, "Deployment issues for multi-user audio support in CVEs", in *Proc. of the ACM Symposium on Virtual Reality Software and Technology*, pp. 179-185, 2002.
- [11] C. Perkins, O. Hodson, V. Hardman, "A survey of packet loss recovery techniques for streaming audio", in *IEEE Network Magazine vol. 12*, pp. 40-48, 1998.
- [12] O. Hodson, C. Perkins, V. Hardman, "Skew detection and compensation for internet audio applications", in *Proc. of the IEEE International Conference on Multimedia and Expo*, 2000.
- [13] Speex, <http://www.speex.org>
- [14] N. Sebe, M. Lew, N. Huisman, "Toward Improved Ranking Metrics", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1132-1143, 2000.
- [15] ITU-T Recommendation G.114, "One-Way Transmission Time", <http://www.itu.int/rec/T-REC-G.114>, May 2003.