

Faculty of Informatics

Faculty of Informatics - Papers

University of Wollongong

Year 2005

Rendering Models for Immersive Voice Communications within Distributed Virtual Environment

Y. P. Que*

P. Boustead†

F. Safaei‡

*University of Wollongong

†University of Wollongong, boustead@uow.edu.au

‡University of Wollongong, farzad@uow.edu.au

This article was originally published as: Que, YP, Boustead, P & Safaei, F, Rendering Models for Immersive Voice Communications within Distributed Virtual Environment, IEEE International Region 10 Conference (TENCON 2005), Melbourne, November 21-24 2005, 1-6. Copyright 2005 IEEE.

This paper is posted at Research Online.

<http://ro.uow.edu.au/infopapers/500>

Rendering Models for Immersive Voice Communications within Distributed Virtual Environment

Ying Peng Que, Paul Boustead, Farzad Safaei
Smart Internet CRC,

Telecommunications and Information Technology Research Institute,
University of Wollongong, Australia

Email {Ying, farzad, paul}@titr.uow.edu.au

Abstract—This paper compares three possible rendering models for the provision of Immersive Voice Communications (IVCs) in Distributed Virtual Environments (DVEs) such as multiplayer online games. The common aim of these three rendering models is to create a personalised auditory scene for each *listening avatar*, consisting of a mix of the surrounding avatars' voices, positioned according to their positions in the virtual world. The first two rendering models are based on amplitude panning localisation and HRTF-based binaural localisation respectively. The computation cost of the latter is deemed too large to meet the identified processing power constraints. A computation reuse scheme was introduced in the third rendering model which, as shown in our simulation results, reduces significantly the computational cost of providing IVC using HRTF-based binaural localisation.

I. INTRODUCTION

In recent time, Distributed Virtual Environment (DVE) has seen many applications over the Internet [1] [2] [3]. DVE is also known as Networked Virtual Environment (NVE) or Collaborative Virtual Environments (CVE). A DVE is a virtual environment that is distributed over a common underlying network. Each DVE user is represented by an avatar in the virtual world. Multiple DVE users from different locations in the physical world explore the virtual world together and interact with each other [1]. One typical example of DVE is Multi-player Online Games (MOG) such as Lineage which had 3.8 million subscribers in 2003 [4].

In [5], the notion of immersion in a virtual environment is defined as the sense of being surrounded by the stimuli of the virtual environment. Previously, virtual environment developers have placed greater emphasis on the visual stimuli for creating immersion. However, there are empirical results suggesting that the perceived quality of the visual displays can be improved when presented in conjunction with either medium or high quality sound [6]. Moreover, within the virtual space of DVE, avatars demand close interactions and co-operations. Thus, a real-time multipoint-to-multipoint Immersive Voice Communication (IVC) system could be considered very useful for a DVE. The IVC rendering model creates a personalised auditory scene for each *listening avatar* which consists of a mix of all the *speaking avatar* voices within its hearing range, each rendered with cues for their respective direction and distance. The number of avatars within a DVE can be large. More importantly, these avatars can be very close in the virtual space but yet spread over a large geographical scale in the physical world. It is therefore important for the IVC rendering model to be

well balanced between its rendering quality and scalability. Scalability in this context is measured in terms of the model's efficiency in its usage of the available computational and bandwidth resources. One of the most common bandwidth bottlenecks faced by the DVE network is the access bandwidth of different client platforms. On the other hand, less powerful processing platforms such as legacy systems, and more importantly, the emerging generation of mobile and handheld devices, create computation power bottlenecks. None of the prior art systems reviewed thus far has satisfactorily addressed these bottlenecks for supporting true IVC among a large number of avatars, especially when the distribution of avatars is highly dense. There are some high fidelity immersive audio rendering systems such as that described in [7] which can render multiple sound sources but not on a distributed basis for multipoint-to-multipoint communications. Such systems often attempt to simulate sophisticated room acoustical effects which are too computationally expensive to be applied on a distributed basis for supporting IVC [7]. On the other hand, the current networked voice communication systems are either text-based [8] or simple mono Voice over IP (VoIP) applications [9], neither of which can really provide a sense of immersion.

In this paper, we first examine a few possible architectures for delivering IVC for DVE and identify the reasons and likely scenarios for using a delivery architecture which places the computational load mainly on dedicated servers. We then compare three possible types of IVC rendering models based on this delivery architecture. A novel computation reuse scheme is introduced to reduce the high computation complexity problem incurred by the basic HRTF-based rendering model. This computation reuse scheme is based on the concept of *acceptable angular error* which, in essence, trades off rendering accuracy for computational complexity by prioritising the rendering accuracies of the *listening avatars* according to their distances away from the *speaking avatar*.

The rest of this paper is organised as follows: Section II examines a few potential delivery architectures for the IVC rendering model and provides an overview of the three possible types of IVC rendering models. A series of simulation results are presented in Section III, evaluating the respective scalabilities of the three IVC rendering models. Finally the conclusion is drawn in Section VI.

II. SYSTEM OVERVIEW

A. The IVC Delivery Architecture

A few types of delivery architectures have been previously considered for the efficient transmission of live voice streams for the IVC system [1] [2]. The first one is the peer-to-peer architecture where each *listening avatar* receives the mono (un-rendered) voice streams from all other avatars within its hearing range. The auditory scene creation then takes place locally at the access platforms of the DVE users. This architecture offers the best delay performance for IVC and utilises the existing “*free*” (at no cost to the service provider) processing power of the user platforms [1]. However, the peer-to-peer architecture consumes large amount of access bandwidth as well as core network bandwidth. In the peer-to-peer architecture, a *listening avatar’s* access platform must download one mono voice stream for each *speaking avatar* within the *listening avatar’s* hearing zone (the circle region with the *listening avatar* at the centre and its hearing range as the radius). Similarly, each *speaking avatar’s* access platform must upload one mono voice stream to each *listening avatar* within the *speaking avatar’s* audible zone (the circle region with the *speaking avatar* at the centre and its audible range as the radius). In addition, the peer-to-peer architecture has other limitations, most noticeably, the privacy and security problems associated with users having to send voice streams directly to each other [1].

The second type of delivery architecture is the server-client architecture in which dedicated servers forward voice streams among clients. Such use of dedicated servers overcomes the aforementioned privacy and security issues in the peer-to-peer architecture. More importantly, the use of dedicated servers also reduces the access bandwidth required to support the IVC system.

An example of delivering IVC over server-client architecture is the Dense Immersive Communication Environment (DICE) system described in [10]. In the DICE architecture, the access platforms still perform the auditory scene creation locally but with the assistance of dedicated servers. DICE retains the peer-to-peer approach’s advantage in exploiting the “*free*” client processing power while reducing the access bandwidth required by each client. This access bandwidth usage reduction is achieved because each client only needs to send a single mono voice stream to the corresponding server which in turn, transmits only K cluster streams to each client. Each cluster stream is a weighted mix of the individual voice streams in a segment of the auditory scene. K is limited by the available access bandwidth and is set to be a small value, e.g. 4 in [10].

In this work, we adopt another type of server-client architecture which differs from the DICE-like architecture. Our architecture is server-centric in the sense that the auditory scene creations are carried out centrally at the servers so as to minimise the computational load on the clients. While DICE caters well for a wired network with high performance access platforms, our architecture is better suited to deliver IVC over mobile and wireless devices, which might not be able to create auditory scenes locally due to limited access bandwidth, low

computational power available and above all, the battery power constraint. Similar to DICE, our architecture offers much better access bandwidth efficiency than the peer-to-peer architecture. Each client in our architecture sends a single mono stream to the corresponding server. The sever sends only C mixed rendered streams back to each client for final playback. C denotes the number of output channels per auditory scene creation as entailed by the localisation technique. C is small for both of the two localisation techniques examined in this work. The access bandwidth efficiency of our architecture actually surpasses that of the DICE architecture when applying the HRTF localisation technique with C of only 2 (see C.2 of Section II). If the DICE architecture is to match this performance by setting K to 2, the angular and distance error would be too great to justify (K was actually chosen to be 4) [10]. A major downside of our architecture is that, compared to DICE and the peer-to-peer architectures, more powerful servers are required in our architecture to carry out the auditory scene creations centrally. In order to cover the expensive costs of using high power servers, the users of our IVC system, e.g. mobile gamers, might have to pay a higher access fee than that charged by the IVC systems delivered over other architectures.

B. The auditory scene creation process

In the context of this work, we use the term *vector* to describe the direction and distance of the rendered voice of a particular *speaking avatar* with respect to a particular *listening avatar*. For example, in Fig. 3, the *vector* $\overline{A_s A_{L1}}$ refers to the rendered voice of the *speaking avatar* A_s with respect to the listening avatar A_{L1} . The auditory scene creation process consists of two stages. The first stage is the vector positioning operation which localises a *vector* by defining its direction and then adds the perception of distance to the localised *vectors*. Due to the computational power constraint, the distance perception is currently created through a simple amplitude weighting operation according to the inverse square law of sound propagation through free-space [11]. However, sophisticated models of reflections and reverberations [16] can be incorporated into our IVC system if the processing power limit permits. In the second stage of scene creation, all the rendered streams belonging to the same output channel are linearly mixed into one stream. The number of linear mixing operations is given by

$$D \times S - C. \quad (1)$$

We define the *avatar density* (D) as the average number of *speaking avatars* per auditory scene or hearing zone of any given *listening avatar*. S denotes the number of rendered voice streams per input voice stream. Fig. 1 illustrates a simple example where there is initially 4 mono voice streams in a hearing zone ($D = 4$). After rendering each mono voice stream using Head Related Transfer Function (HRTF) with S of 2 (see C.1 of Section II), 8 rendered streams are produced. Then all the rendered voice streams corresponding to the same output channel are mixed together. In the case of HRTF localisation with a C of 2, two final mixed streams are produced for final playback (see C.1 of Section II).

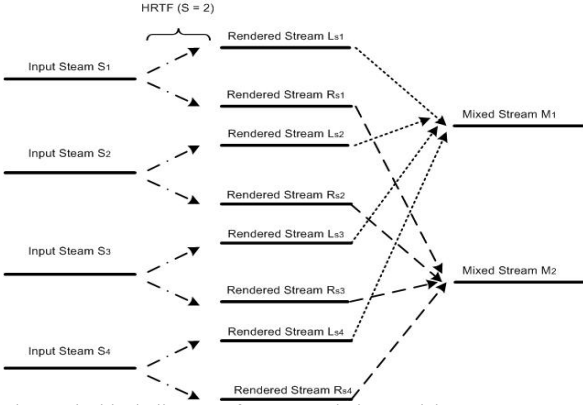


Fig. 1. The block diagram of *HRTF* rendering model

C. Three Rendering Models for IVC

For the current scope, we limit the IVC rendering model to Two-Dimensional (2-D), i.e. azimuth only. However, the two sound localisation techniques investigated herein can be both extended to include elevation, thus upgrading the IVC rendering models to Three-Dimensional (3-D).

C.1 “Amplitude Panning” rendering model

The *Amplitude Panning* rendering model is based on the amplitude panning localisation technique which is formulated in vector notation by Pulkki [12]. In its simplest configuration, amplitude panning applies two coherent signals ($S=2$) to a pair of speakers, each derived via a different gain adjustment from the input sound source. The amplitude difference between the two coherent signals creates the perception of a virtual sound source localised on an active region between the two loudspeakers. However, amplitude panning can never localise a virtual source outside the active region. In order to provide a 360° 2-D listening area via amplitude panning, we choose the widely-accepted five speaker configuration ($C=5$) used by the international 5.1 channel surround sound standard [12].

The real-time computational cost of 2-D amplitude panning is one Floating Point Operations (FLOPs) per output sample, attributed to the real multiplications of the input sound signal with the two gain factors. Despite offering such low computational cost, the 5 speaker playback system creates a portability problem as it is space-consuming and inconvenient to relocate and set up and is certainly not suited to provide IVC for mobile DVE users.

Amplitude Panning performs the localisation and distance weighting of a vector in one step as the gain factor used in amplitude panning also controls the amplitude of the localised sound source. Let N denote the length of input voice stream. The average positioning cost per vector (SPC_{vec}) of the *Amplitude Panning* model is,

$$S \times N = 2N \text{ FLOPs.} \quad (2)$$

In the “*Amplitude Panning*” rendering model, only real signals are processed. Thus applying (1), the average linear mixing cost per auditory scene (LMC_{scene}) for this model is

$$(2D - 5) \times N \text{ FLOPs.} \quad (3)$$

The average overall rendering cost per auditory scene (RC_{scene}) for this model is,

$$SPC_{vec} \times D + LMC_{scene} = 2N \times D + (2D - 5) \times N \text{ FLOPs.} \quad (4)$$

C.2 “HRTF” rendering model

The second type of IVC rendering model employs a binaural localisation technique which uses the Head Related Transfer Function (HRTF) [13]. HRTF is measured and stored as the Head Related Impulse Response (HRIR) which is the time domain representation of HRTF. For each defined sound source direction, the two corresponding HRIR (left and right ears respectively) are retrieved from the database [13] and then convolved (binaural synthesis) separately with the original sound source ($S=2$) before being played back on the user’s headset ($C=2$). The HRTF filter used in this work is the compact filter with an order of 128 [13]. Because we only process voice in short 20 ms frames, we apply the short sequence version of the Fast Fourier Transform (FFT) based fast circular convolution [14] to our binaural synthesis. In order to avoid overlapping problem in circular convolution, we pad the input sequence length N to 1024. This N value is also applied to the *Amplitude Panning* model. Hence, the average positioning cost per vector (SPC_{vec}) of the *HRTF* rendering model is

$$107 \times S \times N = 107 \times 2N \text{ FLOPs.} \quad (5)$$

Fig. 2 illustrates the creation of a simple auditory scene with three vectors using this FFT-based binaural localisation.

Equation (5) states that the SPC_{vec} of the *HRTF* rendering model is 107 times higher than the *Amplitude Panning* rendering model. Such high computational cost is due to the fact that the *HRTF* rendering model applies binaural localisation to position all the vectors in a DVE according to their exact positions.

Despite of its high computational cost, the *HRTF* rendering model offers two advantages over the *Amplitude Panning* model. Firstly, the *HRTF* rendering model offers lower access bandwidth consumption than the *Amplitude Panning* model. The server-centric delivery architecture chosen for our work has an average access bandwidth of $C+1$ per avatar per auditory scene. The *HRTF model* has a C of 2 whereas the “*Amplitude Panning*” model has a C of 5. Secondly, binaural localisation is optimised for playback on headphones which is less complex and more portable than the 5 speaker playback system required by the amplitude panning approach. These advantages justify the need to develop a mechanism to reduce the computational complexity of the *HRTF* rendering model.

C.3 “Computation Reuse” model

In this work, we propose a *Computation Reuse* rendering model which only applies accurate binaural localisation to a small percentage of vectors located close to a given *speaking avatar*, while performing less accurate localisation for the other vectors located further away. This reuse scheme is based on the concept of “acceptable angular error” ($\epsilon_{acceptable}$) which refers to the acceptable level of angular deviation between the perceived

position and the exact position of a sound source in the virtual world. This concept of *acceptable angular error* was first proposed in [10] and implemented for a cluster-based computation reduction scheme. In [10], $\epsilon_{\text{acceptable}}$ is assumed to increase linearly with the maximum distance between a particular pair of *listening avatars* and the *speaking avatar*. A similar relationship is used in this work. To prevent extreme values of angular errors, the *angular-distance relationship* is bounded with two angles, a maximal value ($\epsilon_{\text{max}}=30^\circ$) and a minimum value ($\epsilon_{\text{min}}=0.1\epsilon_{\text{max}}$). It is worth noting that the angular values we used here are more conservative (thus offering higher rendering accuracy) than in [10] which uses the values of $\epsilon_{\text{min}}=15^\circ$ and $\epsilon_{\text{max}}=45^\circ$. Let d denotes the maximum distance between a particular pair of *listening avatars* and the *speaking avatar*. Let r_{mean} denotes the mean audible range of all the *listening avatars*. For $d < r_{\text{mean}}$,

$$\epsilon_{\text{acceptable}} = \epsilon_{\text{min}} + (\epsilon_{\text{max}} - \epsilon_{\text{min}}) \times \frac{d}{r_{\text{mean}}} = 0.1\epsilon_{\text{max}} + 0.9\epsilon_{\text{max}} \times \frac{d}{r_{\text{mean}}}, \quad (6)$$

When $d \geq r_{\text{mean}}$, $\epsilon_{\text{acceptable}}$ is set to a maximal value of ϵ_{max} ,

$$\epsilon_{\text{acceptable}} = \epsilon_{\text{max}}. \quad (7)$$

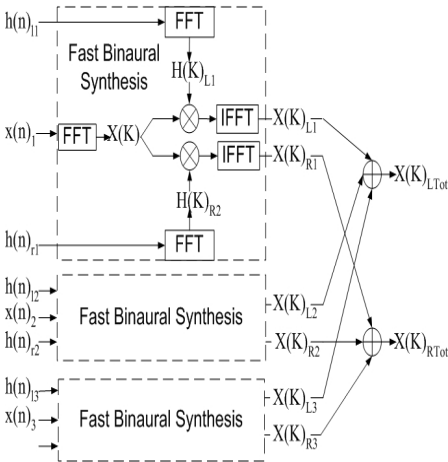


Fig. 2. The block diagram of HRTF rendering model

Although the FFT-based fast convolution generates intermediate complex signals, the last Inverse Fourier Transform (IFFT) step transforms the final resultant signals back to the real domain. Hence only real signals are processed in the distance weighting operation and the linear mixing operation of this model. Applying (1), the common the average linear mixing cost per auditory scene (LMC_{scene}) for both HRTF and *Computation Reuse* rendering models is $(2D - 2) \times N$

FLOPs per scene. (9)

Consequently, the average overall rendering cost per auditory scene (RC_{scene}) of this rendering model is,

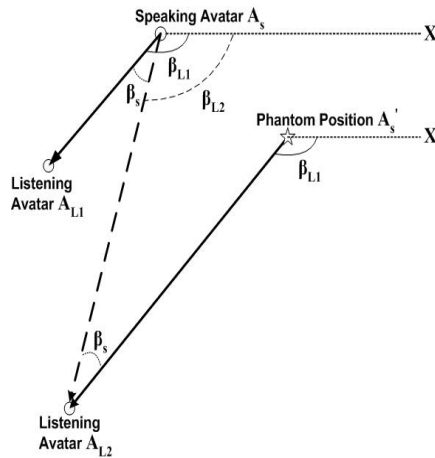


Fig. 3. Computation reuse between two vectors

To illustrate the operation of our computational reuse algorithm, we study the computational reuse between two adjacent vectors, $\overrightarrow{A_s A_{L1}}$ and $\overrightarrow{A_s A_{L2}}$ with respect to *speaking avatar* A_s . As shown in Fig. 3, the angular spread between the two vectors is given by $\beta_{L1} - \beta_{L2} = \beta_s$. If $\beta_s \leq \epsilon_{\text{acceptable}}$ of the *listening avatar* A_{L2} (further away from A_s), instead of being rendered to its exact position, A_{L2} receives a distance weighted version of the rendering results produced for the other *listening avatar* A_{L1} (closer to A_s). From the perspective of A_{L2} , this computation reuse creates the perception of the voice of A_s emanating from the phantom position A_s' , deviating from the exact position of A_s by β_s . It must be noted it is assumed in the scenario of Fig.3, both of the listening avatars have the same facing orientation.

The computational reuse level (L_{reuse}) for the “*Computation Reuse*” rendering model is defined as the percentage of the total number of vectors in the DVE that can reuse the rendering computations of the other vectors. The computational cost of reusing another vector’s rendering result is merely the distance weighting cost of 1 FLOPs per output sample. Hence, the average vector positioning cost per vector (SPC_{vec}) of the *Computation Reuse* rendering model is

$$107 \times (1 - L_{\text{reuse}}) \times 2N + (1 \times L_{\text{reuse}}) \times 2N \quad (8)$$

FLOPs per vector.

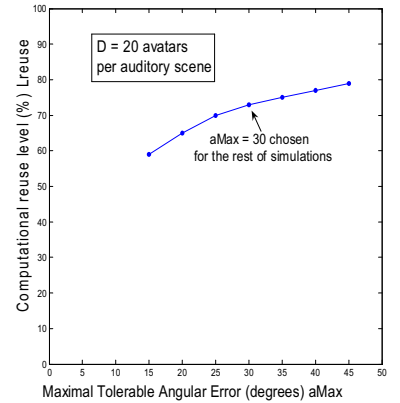


Fig. 4. The effect of varying acceptable angular error on rendering computation reuse level

$$(107 \times (1 - L_{\text{reuse}}) \times 2N + (L_{\text{reuse}}) \times 2N) \times D + (2D - 2) \times N \quad (10)$$

FLOPs per scene.

D. Heuristics algorithm

A heuristics algorithm was developed in this work which determines the computational reuse status of each vector in a DVE. Because a particular vector can only reuse the rendering results of another vector in the same audible zone (from the same *Speaking Avatar* voice), this heuristics is performed separately for the audible zone of each *Speaking Avatar*.

Variables:

A_t : the total number of audible zones to be processed in the DVE.

A_p : the count of audible zones processed.

$L_1^{A_p}$: the local list of vectors to be processed for the current (A_p^{th}) audible zone.

$L_2^{A_p}$: the local list of vectors to be exactly rendered in the A_p^{th} audible zone.

$sortL_1^{A_p}$: $L_1^{A_p}$ sorted in descending order according to the vectors' distances away from the *speaking avatar*'s position.

$L_1^{A_p}(1)$: the next vector to be processed which is always the first one in $L_1^{A_p}$.

V_e : the first vector V_e that can be *acceptably reused* by $L_1^{A_p}(1)$.

Pseudo code:

```

 $A_p = 1$ ;
While  $A_p \leq A_t$ ,
   $L_1^{A_p} =$  all vectors  $\in A_p^{th}$  audible zone,
   $sortL_1^{A_p} = \text{Sort}(L_1)$ ,  $L_2^{A_p} = 0$ ;
  While  $\text{length}(L_1) > 0$ 
    If  $\text{length}(L_2^{A_p}) > 0$ 
      Iterate  $L_2^{A_p}$ , search for  $V_e$ 
      If find
        delete  $L_1^{A_p}(1)$  and  $V_e$  from  $L_1^{A_p}$ , add  $V_e$  to  $L_2^{A_p}$ ,
      continue;
    end
  end
  Iterate  $L_1^{A_p}$ , search for  $V_e$ .
  If find
    delete  $L_1^{A_p}(1)$  and  $V_e$  from  $L_1^{A_p}$ , add  $V_e$  to  $L_2^{A_p}$ ,
  continue;
  end
End
 $A_p = A_p + 1$ ;
End

```

III. SIMULATIONS

A. Simulation Set Up

In our simulation, we use uniform random distribution for the position of avatars in the virtual world. We study the performance of the three IVC rendering models at one particular time instant over our server-centric architecture. In order to vary the *avatar density* of the virtual world, we kept the total number of avatars in the virtual world at 400 and varied the size of the virtual world, which was modelled as a square area. We also assume all the avatars can hear but only half of them are speaking at any given time with a fixed audible range of 30 m.

B. The Effect of Varying Acceptable Angular Error on Computational Reuse Level

As shown by Fig. 3, at a given distance away from speaking avatar (d), a larger value of $\epsilon_{\text{acceptable}}$ increases the likelihood of computation reuse (measured by L_{reuse}) between a given pair of adjacent *vectors*. Moreover, at a given distance, as set in

Eqn. (6) and (7), the value of $\epsilon_{\text{acceptable}}$ depends on the maximal acceptable angular value ϵ_{max} . Hence we can study the effect of varying $\epsilon_{\text{acceptable}}$ on L_{reuse} by observing Fig. 4 which shows L_{reuse} vs increasing ϵ_{max} . As shown by Fig. 4, at a fixed *avatar density* of $D=20$ avatars per auditory scene, for an increase of 40° in ϵ_{max} from 15° to 45° , the corresponding increase in L_{reuse} is only 20 %. In order to avoid unnecessarily large angular error at large distance away, we choose the middle value of $\epsilon_{\text{max}} = 30^\circ$. It should be noted that there has been no conclusive study on what is acceptable and 30 degrees has been chosen to further explore the performance of the reuse mechanism in the rest of our simulations.

C. The Effect of Rising Avatar Density on Computational Reuse Level

It can be seen in Fig. 5 that the computational reuse level L_{reuse} increases with rising *avatar density* D . This is because of fact that as D rises, the avatars are more densely populated with smaller angular and geometric distance separations, leading to more cases of computation reuse.

D. Computational Cost Comparison

Fig. 6 shows that the average positioning costs per vector (SPC_{vec}) of the *Computation Reuse* model decreases with rising avatar densities. This can be explained by combing Eqn. (8) with the trend shown in Fig. 5. Equation (8) states that the SPC_{vec} values of the *Computation Reuse* model decreases with increasing computational reuse level L_{reuse} and Fig. 5 shows that the L_{reuse} values of the *Computation Reuse* model increases with rising avatar densities.

Fig. 7 compares the three rendering models in terms of their average overall rendering cost per auditory scene (RC_{scene}). As stated in Eqn. (4), RC_{scene} is the sum of $SPC_{\text{vec}} \times D$ and LMC_{scene} . The value of the latter is rather insignificant compared to the former for the HRTF and computational reuse methods. Therefore the trends observed on Fig. 7 can be explained by studying the effects of rising avatar densities on the SPC_{vec} values of the three rendering models. The RC_{scene} plots of both the *HRTF* and *Amplitude Panning* models are linearly increasing. This is consistent with Eqns. (2) and (5). On the other hand, the RC_{scene} curve of the *Computation Reuse* model lies in between these two straight lines. Beyond the avatar densities of 10 avatars per hearing zone, the gap between the RC_{scene} plots of the *Computation Reuse* model and the *Amplitude Panning* model is consistently much smaller than that between the *Computation Reuse* model and the *HRTF* rendering model. More importantly, the rate of increase of the RC_{scene} curve for the *Computation Reuse* model decreases as D rises. Such trend is due to the fact that the SPC_{vec} values of the *Computational Reuse* model decrease with rising avatar densities as shown by Fig. 6 previously.

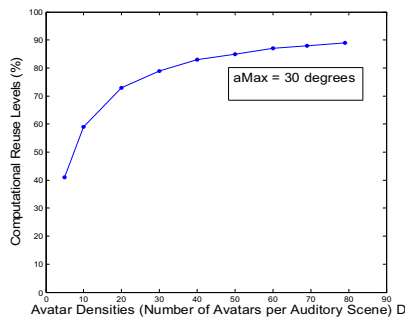


Fig. 5. The effect of avatar densities on computational reuse level

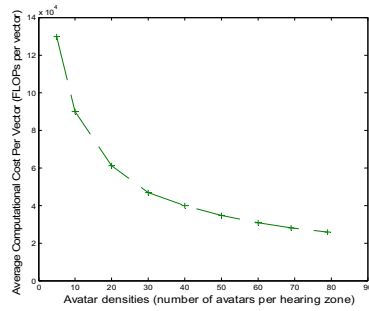


Fig. 6. Average vector positioning cost of Computational Reuse

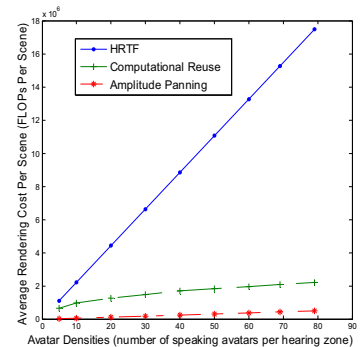


Fig. 7. Comparing the average rendering computational cost per auditory scene

E. Summary of Results and Recommendation

Of the three rendering models studied, the *Amplitude panning* rendering model offers the lowest overall computational complexity. The basic *HRTF* rendering model incurs much higher overall computational complexity than the other two models. For medium to high avatar densities, the *Computation Reuse* rendering model offers HRTF-based vector localisation but at a significantly reduced computational cost than the basic *HRTF* rendering model.

Due to the problem of hardware scalability and portability associated with using multiple speakers, the "*Amplitude panning*" model is not suited to provide IVC for the mobile DVE applications targeted by our server-centric delivery architecture. Although the basic *HRTF* rendering model is more accurate (all vectors are rendered to their exact positions) than the *Computation Reuse* model, it is highly computationally intensive and thus is also not suited to the mobile DVE applications which can accept some angular positioning error. The *Computation Reuse* model offers HRTF-based vector localisation at reasonable computational costs. This model is well suited to mobile DVE application scenarios with stringent constraints on access bandwidth, battery power supply and computational resources.

IV. CONCLUSIONS

The main contribution from this work is the introduction of the computation reuse scheme based on a concept of *acceptable angular error*. At medium to high *avatar densities*, this scheme significantly reduces the computational complexity of the basic *HRTF* rendering model and approaches the best case *Amplitude Panning* model. Of the three rendering models studied, only the *Computation Reuse* model can satisfy the constraints imposed by mobile DVE applications. In addition, the computational costs of the rendering models were measured realistically in FLOPs, based on two well-know localisation techniques, which should provide useful insights to any future implementations of immersive voice communication service.

ACKNOWLEDGEMENT

This work is supported by the Co-operative Research Centre for Smart Internet CRC (SITCRC) and the University of Wollongong (UOW), Australia.

REFERENCES

- [1] Paul Boustead and Farzad Safaei, "Comparison of Delivery Architectures for Immersive Audio in Crowded Networked Games", in *Proc. of the 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Ireland, 16-18 June, 2004.
- [2] J. Bolot and F. Parisi, "Adding Voice to Distributed Games on the Internet", in *Proc. of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 480-487, 1998.
- [3] M. Radenkovic, C. Greenhalgh, S. Benford, "Deployment issues for multi-user audio support in CVEs," in *ACM Symposium on Virtual Reality Software and Technology*, 2002, pp. 179-185.
- [4] Rajiv Mehrotra, "South Korea beats the world in broadband gaming", *IEEE Multimedia*, Vol. 10, Iss. 2, pp. 12-14, Apr.-Jun. 2003.
- [5] B. Witmer and M. Singer, "Measuring presence in virtual environments: A Presence Questionnaire", *Presence: Teleoperators and Virtual Environments*, Vol. 7, No. 3, 225-240, 1998.
- [6] D. R. Begault, R. E. Ellis, and E. M. Wenzel, "Headphone and Head-mounted Visual Displays for Virtual Environments", in *Proc. of AES 15th International Conference*, Copenhagen, Denmark, October 31 - November 2, 1998, pp. 213-217.
- [7] Martin Naef, Oliver Staadt, Marjus Gross, "Spatialised Audio Rendering for Immersive Virtual Environments", in *Proc. of the ACM symposium on Virtual reality software and technology*, pp 65-72, 2002.
- [8] ICQ Inc, ICQ Home Page, <http://www/icq.com>, (18 Dec. 2004).
- [9] TeamSpeak, The Team Pay Engine, <http://www.teamspeak.org>, (18 Dec. 2004).
- [10] Paul Boustead, Farzad Safaei and Mehran Dowlatshahi, "DICE: Internet Delivery of Immersive Voice Communication for Crowded", to appear in *Proc. of IEEE Virtual Reality (VR) 2005*, Bonn, Germany, Mar. 12-16, 2005.
- [11] David M. Howard and James Angus, *Acoustics and Psychoacoustics (Music Technology Series)*, Focal Press, Oxford, U.K, 2001.
- [12] Thomas Funkhouser, Patrick Min and Ingrid Carlbom, "Real-Time Acoustic Modelling for Distributed Virtual Environments", in *Proc. of the 26th annual conference on Computer graphics and interactive techniques*, pp. 365-374, 1999.
- [13] Ville Pulkki, "Virtual Sound Source Positioning Using Vector Base Amplitude Panning", *J. Audio Eng. Soc.*, vol. 45, no. 6, pp. 456-466, Jun. 1997.
- [14] Durand R. Begault, *3-D Sound for Virtual Reality and Multimedia*, Academic Press Professional, Cambridge, MA, USA, 1994.
- [15] A. Mertins. *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications*, John Wiley & Sons, Chichester, 1999.