Progressive Perceptual Audio Rendering of Complex Scenes

Thomas Moeck[1,2] Nicolas Bonneel[1] Nicolas Tsingos[1] George Drettakis[1] Isabelle Viaud-Delmon David Alloza
[1]REVES/INRIA Sophia-Antipolis CNRS-UPMC UMR 7593 EdenGames
[2]Computer Graphics Group, University of Erlangen-Nuremberg

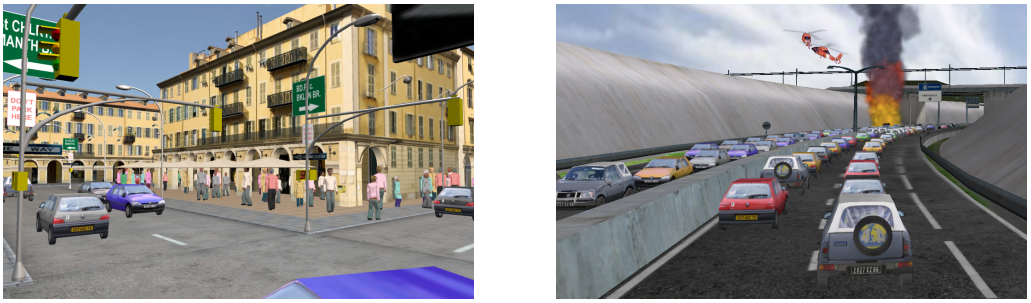

Figure 1: Left: A scene with 1815 mobile sound sources. Audio is rendered in realtime with our progressive lossy processing technique using 15% of the frequency coefficients and with an average of 12 clusters for 3D audio processing. Degradations compared to the reference solution are minimal. Right: a scene with 1004 mobile sound sources, running with 25% of the frequency coefficients and 12 clusters.

Abstract

Despite recent advances, including sound source clustering and perceptual auditory masking, high quality rendering of complex virtual scenes with thousands of sound sources remains a challenge. Two major bottlenecks appear as the scene complexity increases: the cost of clustering itself, and the cost of pre-mixing source signals within each cluster.

In this paper, we first propose an improved hierarchical clustering algorithm that remains efficient for large numbers of sources and clusters while providing progressive refinement capabilities. We then present a lossy pre-mixing method based on a progressive representation of the input audio signals and the perceptual importance of each sound source. Our quality evaluation user tests indicate that the recently introduced audio saliency map is inappropriate for this task. Consequently we propose a "pinnacle", loudness-based metric, which gives the best results for a variety of target computing budgets. We also performed a perceptual pilot study which indicates that in audio-visual environments, it is better to allocate more clusters to visible sound sources. We propose a new clustering metric using this result. As a result of these three solutions, our system can provide high quality rendering of thousands of 3D-sound sources on a "gamer-style" PC.

Keywords: Audio rendering, auditory masking, ventriloquism, clustering

1 Introduction

Spatialized audio rendering is a very important factor for the realism of interactive virtual environments, such as those used in computer games, virtual reality, or driving/flight simulators, etc. The complexity and realism of the scenes used in these applications has increased dramatically over the last few years. The number of objects which produce noise or sounds can become very large, e.g., cars in a street scene, crowds in the stadium of a sports game etc. In addition, recent sophisticated physics engines can be used to synthesize complex sound effects driven by the physics, for example the individual impact sounds of thousands of pieces of a fractured object.

Realistic 3D audio for such complex sounds scenes is beyond the capabilities of even the most recent 3D audio rendering algorithms. The computational bottlenecks are numerous, but can be grouped into two broad types: the cost of *spatialization*, which is related to the audio restitution format used; and the *per sound source* cost, which relates to the different kinds of effects desired. An example of the former bottleneck is Head Related Transfer Function (HRTF) processing for binaural rendering [Møller 1992], or the rendering of numerous output channels for a Wave Field Synthesis (WFS) system [Berkhout et al. 1993], which can use hundreds of speakers. The latter bottleneck includes typical spatial effects such as delays, the Doppler effect, reverberation calculations, but also any kind of studio effect the sound engineers/designers wish to use in the application at hand.

Recent research has proposed solutions to these computational limitations. Perceptual masking with sound source clustering [Tsingos et al. 2004], or other clustering methods [Herder 1999; Wand and Straßer 2004] do resolve some of the issues. However, the clustering algorithms proposed to date are either restricted to static scenes, or add an unacceptable computation overhead due to a quadratic step in cluster construction when the number of sources is large. In addition, the cost of per source computation, sometimes called premixing, can quickly become a bottleneck, again for complex soundscapes.

In this paper we present two contributions:

- The first contribution is the speed improvement of the clustering and premixing steps of the [Tsingos et al. 2004] approach. First, we introduce a new recursive clustering method, which can operate with a fixed or variable target cluster budget, and thus addresses the issue of spatialization cost mentioned above. Second, we develop a novel "pinnacle-based" scalable premixing algorithm, based on [Tsingos 2005], pro-

viding a flexible, perceptually-based framework for the treatment of the per-source computation costs.

- The second contribution is the investigation of perceptual issues related to *clustering* and *premixing*, based on *pilot user studies* we conducted. In particular, we investigate the influence of visuals on audio clustering for audio-visual scenes, and propose a modified metric taking into account the indication that it is probably better to have more sources in the view frustum. For scalable premixing, we evaluated the different metrics which can be used, including recently developed perceptual models, such as the audio saliency map [Kayser et al. 2005], and performed a perceptual quality test for the new algorithm.

In the following, we briefly overview related previous work, including that in the acoustics and perception literature. We then present the new recursive clustering method in Sect. 3, the new study and the resulting algorithm for scalable premixing in Sect. 4 and the study and novel metric for crossmodal audio-visual clustering in Sect. 5. We present some implementation issues and results, then conclude with a discussion of our approach.

2 Previous work

Relatively little effort has been devoted to the design of scalable rendering strategies for 3D audio, which can provide level-of-detail selection and graceful degradation. We give below a short overview of the previous work most relevant to our problem.

Encoding and rendering of spatial auditory cues. Progressive spatial sound encoding techniques can be roughly subdivided in two categories.

A first category is based on a physical reconstruction of the wavefield at the ears of the listener. For instance, several approaches have been proposed to perform progressive binaural rendering [Blauert 1997] by decomposing HRTFs on a set of basis functions through principal component analysis [Chen et al. 1995; Larcher et al. 2000; Jot and Walsh 2006]; while providing level-of-detail, they are limited to this unique restitution format. Alternatively, decomposition of the wavefield can be used (e.g., spherical harmonics) [Malham and Myatt 1995] for level-of-detail, relatively independently of restitution setup (e.g., conversion to binaural format [Jot et al. 1999]). High accuracy requires a large number of channels however, limiting the methods' applicability.

Another category performs world-space compression of positional cues by clustering nearby sound sources and using a unique representative position per cluster for spatial audio processing [Herder 1999; Wand and Straßer 2004; Tsingos et al. 2004]. Wand et al. [Wand and Straßer 2004] group sources in a hierarchical spatial data structure as used for point-based rendering. However, their approach is very efficient only in the case of static sources. Tsingos et al. [Tsingos et al. 2004] recently introduced a clustering algorithm driven by a loudness-weighted geometrical cost-function (see Sec. 3). They also use precomputed descriptors (e.g., loudness and tonality) to sort the sources by decreasing importance and perform a greedy culling operation by evaluating auditory masking at each time-frame of the simulation. Inaudible sources can then be safely discarded. Although this approach was found to perform well for environments containing a few hundred sources, it is unclear that it scales well to larger numbers of sources and clusters due to the cost of the proposed clustering algorithm which, in their case, implies a near quadratic number of evaluations of the cost-function.

Scalable audio processing. Fouad et al. [Fouad et al. 1997] propose a level-of-detail progressive audio rendering approach in the time-domain; by processing every n-*th* sample, artifacts are introduced at low budget levels. Wand and Straßer [Wand and Straßer 2004] introduce an importance sampling strategy using random selection, but ignore the signal properties, thus potentially limiting the applicability of this method.

A family of approaches has been proposed to directly process perceptually-coded audio signals [Lanciani and Schafer 1997; Lanciani and Schafer 1999; Darlington et al. 2002; Touimi 2000; Touimi et al. 2004] yielding faster implementations than a full decode-process-recode cycle. Although they are well suited to distributed applications involving streaming over low-bandwith channels, they require specific coding of the filters and processing. Moreover, they cannot guarantee efficient processing for a mixture of several signals, nor that they would produce an optimal result. Other methods have explored how to extend these approaches by concurrently prioritizing subparts of the original signals to process to guarantee a minimal degradation in the final result [Gallo et al. 2005; Tsingos 2005; Kelly and Tew 2002]. Most of them exploit masking and continuity illusion phenomena [Kelly and Tew 2002] to remove entire frames of the original audio data in the time-domain [Gallo et al. 2005] or, on a finer scale, process only a limited number of frequency-domain Fourier coefficients [Tsingos 2005].

Crossmodal studies. While the primary application of 3D audio rendering techniques is simulation and gaming, no spatial audio rendering work to date evaluates the influence of combined visual and audio restitution on the required quality of the simulation. However, a vast amount of literature in neurosciences suggest that cross-modal effects, such as ventriloquism, might significantly affect 3D audio perception [Hairston et al. 2003; Alais and Burr 2004]. This effect tells us that in presence of visual cues, the location of a sound source is perceived shifted toward the visual cue, up to a certain threshold of spatial congruency. Above this threshold, there is a conflict between the perceived sound location and its visual representation and the ventriloquism effect no longer occurs. The spatial window (or angular threshold) of this effect seems to depend on several factors (e.g., temporal synchronicity between the two channels and perceptual unity of the bimodal event) and can vary from a few degrees [Lewald et al. 2001] up to 15° [Hairston et al. 2003].

3 Optimized Recursive Clustering

In previous work [Tsingos et al. 2004], large numbers of sound sources are dynamically grouped together in clusters, and a single new *representative* point source is created. While this approach allows the treatment of several hundred sound sources on a standard platform, it does incur some computational overhead, which becomes a potential bottleneck as a function of the number of input-sources/target-clusters and the available computational budget. To resolve this limitation, we next present a new recursive clustering algorithm. For improved efficiency and accuracy we propose two recursive algorithms: One with a fixed budget of clusters and one with a variable number of clusters. The fixed budget approach allows us to set a fixed computational cost for 3D audio processing and is also useful to address a fixed number of 3D-audio hardware channels when available for rendering. The variable number of clusters dynamically adjusts the computational cost to obtain the best trade-off between quality and speed in each frame, given a user-specified error threshold.

We also discuss a variant of this algorithm which has been included in the commercially available game "Test Drive Unlimited" by EdenGames/ATARI.

In the method of [Tsingos et al. 2004], sources are grouped together

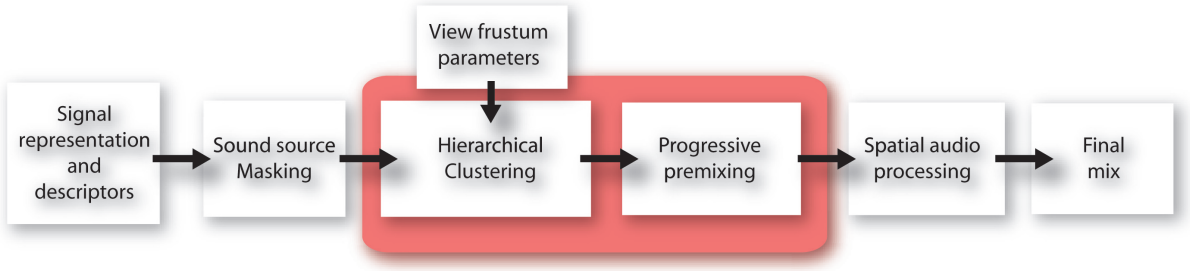

Figure 2: Overview of our overall sound rendering pipeline. In particular, we introduce an improved hierarchical sound source clustering that better handles visible sources and a premixing technique for progressive per-source processing.

by using a clustering algorithm based on the Hochbaum-Shmoys heuristic [Hochbaum and Schmoys 1985]. First, this strategy selects n cluster representatives amongst the k original sources by doing a farthest-first traversal of the point set. The cost-function used is a combination of angular and distance errors to the listener of the candidate representative source with respect to the sources being clustered. An additional weighting term, based on each source's instantaneous loudness value, is used to limit error for loud sources. For details see [Tsingos et al. 2004].

3.1 Recursive fixed-budget approach

In a first pass, we run the original clustering algorithm with a target number of clusters n_0. In each subsequent pass, every generated cluster gets divided into n_k clusters. The total budget of clustering is thus $\prod_k n_k$. The original clustering algorithm can be considered as a special case where only n_0 is set. In our tests, we typically used a two-level recursion.

3.2 Variable cluster budget

This approach dynamically allocates the number of clusters in realtime. This is especially useful for scenes where sounds are frequently changing during time in shape, energy as well as in location. The algorithm then flexibly allocates the required number of clusters; thus clusters are not wasted where they are not needed.

First, every sound source which has not been masked [Tsingos et al. 2004], is put in one cluster which is then recursively split into two until an appropriate condition is met. In every recursion step the error in angle relative to the listener position is computed, for each sound source in a cluster, relative to its centroid. If the average angle error is below a threshold, cluster splitting is terminated. In our tests, we found that a 25° threshold value proved satisfactory.

3.3 Quantitative error/performance comparison

We have performed a quantitative comparison between the previous clustering approach of [Tsingos et al. 2004], and the two proposed recursive clustering methods. We ran several thousand tests with random configurations of 800 sound sources, using the different algorithms and measuring the time and the error (in terms of distance and angle) for each method.

Figure 3 shows the results of the tests for fixed budgets of 12 and 6 clusters using different two-level subdivision strategies. For instance, line 3/4 corresponds to a clustering using 12 clusters (3 top-level clusters recursively refined into 4). The line 12/1 corresponds

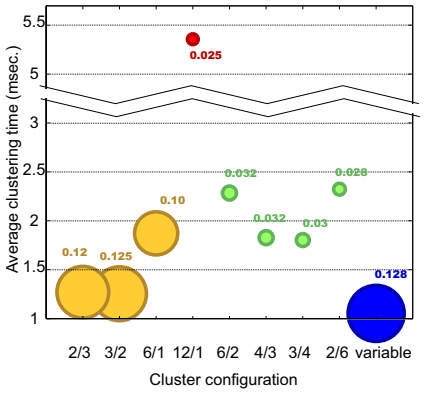

Figure 3: Benchmarks for hierarchical clustering of 800 sources using different 6 and 12 cluster configurations. We display average clustering error (also denoted by circle size). Note the significant speed-up compared to the non-hierarchical algorithm for the 12 cluster configurations (in red) while the errors remain similar.

to the previous clustering approach of [Tsingos et al. 2004] where all 12 clusters are top level.

As we can see, the performance of the recursive approaches are clearly better than the direct algorithm. For the same final budget, the 3/4 and 4/3 configurations appear to be better choices in terms of speed. As expected the error is larger for hierarchical clustering since it leads to less optimal cluster placement. However, as the number of clusters grows this effect tends to disappear. The variable cluster method is faster on average. However, with our current settings it also created fewer clusters (6.6 clusters created on average) and, as a consequence, has higher average error. Interestingly, the peak number of clusters created by the variable method is 22, which underlines the flexibility and adaptability of the approach.

To limit the error compared to a non-hierarchical implementation, it is preferable to create more top level clusters. For instance a 4/3 split is better than a 2/6 split for a 12 cluster configuration, altough it might be slightly slower. Hence, this choice depends on the desired time vs. error tradeoff for the application at hand.

3.4 Implementation in a commercial game

The audio clustering technique was used in the development of the commercially available computer game *Test Drive Unlimited*. In this car racing game, the sound of each racing vehicule is synthesized based on numerous mechanical quantities. The sound emitted by each wheel is controlled by 20 physical variables while 8 variables control the engine/transmission sounds. Four additional vari-

Figure 4: Sound source clustering in the *Test Drive Unlimited* engine. Red wireframe spheres are clusters. ©Eden Games-ATARI 2006.

ables control aerodynamic phenomena. These variables are used for real-time control and playback of a set of pre-recorded sound samples. All sound sources are then rendered in 5.1 or 7.1 surround sound. For implementation on the *XBOX360*, Eden Games adopted a variant of the recursive variable budget technique described above. In particular, a budget of 8 clusters was used at each recursion level. If the quality criterion is not met for this budget, a local clustering step is applied in each cluster. However, the local nature of clustering resulted in audible artifacts from one frame to the next, because of sources moving from one cluster to another. To resolve this problem, sources are ordered by perceptual priority, and the most important ones will prefer to be clustered with the cluster of the previous frame, effecting a form of temporal coherence.

Despite this improvement, extreme cases still presented some difficulty, notably the case of a car crashing into an obstacle. In this case, the physics engine generates numerous short-lasting sound sources in the immediate neighbourhood of the vehicle. Temporal coherence is thus useless in this context. The solution to this issue is to apply a separate clustering step to the sources generated by physics events; this results in more clusters overall, but resolves the problems of quality.

A snapshot of *Test Drive Unlimited* with a visualisation of the sound source clusters superimposed in red, is shown in Figure 4.

4 Scalable perceptual premixing

In order to apply the final spatial audio processing to each cluster (see Figure 2), the signals corresponding to each source must first be *premixed*. The premixing stage can be as simple as summing-up the signals of the different sources in each cluster. In addition, a number of audio effects usually have to be applied on a per-source basis. Such effects include filtering (e.g., distance, occlusions), pitch-shifting (e.g., Doppler effect) or other studio-like effects [Zölzer 2002]. Hence, when large numbers of sound sources are still present in the scene after auditory culling (masking) or for systems with limited processing power, the cost of this stage can quickly become the bottleneck of the audio rendering pipeline.

In this section, we propose a progressive signal processing technique that can be used to implement scalable per-source processing. Our work is based on the approach of [Tsingos 2005] which we briefly summarize in Figure 5.

This approach uses a specific time-frequency representation of audio signals. At each time-frame (typically 1024 samples at 44.1KHz), the complex-valued coefficients of a short-time Fourier transform (STFT) are precomputed and stored in decreasing modulus order. In real-time during the simulation, the algorithm prioritizes the signals and allocates to each source a number of coefficients to process, so that a predefined budget of operations is

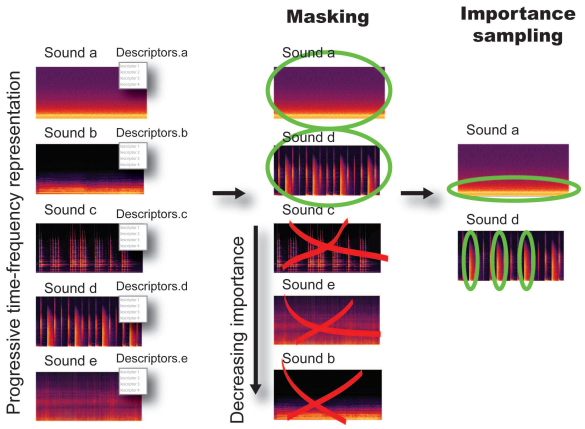

Figure 5: Overview of our progressive perceptual premixing.

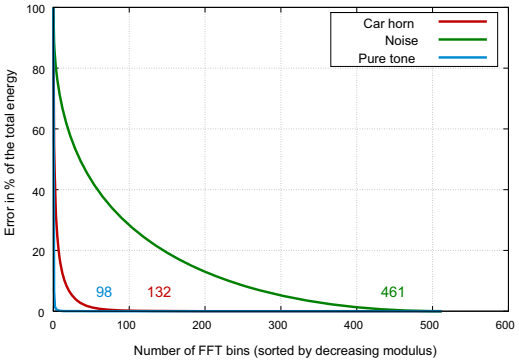

Figure 6: Reconstruction error as a function of target FFT bins. Tonal signals require fewer coefficient than noisier signals since their Fourier representation is much sparser. The value next to each curve corresponds to our pinnacle value. In [Tsingos 2005], the integral of these curves is used to measure the coding efficiency for each frame of input signal.

respected. In [Tsingos 2005], this importance sampling stage is driven by the energy of each source at each time-frame and used to determine the cut-off point in the list of STFT coefficients. However, using only energy for importance sampling leads to sub-optimal results since it does not account for the sparseness of the representation obtained for each signal. For instance, a loud tonal signal might require fewer coefficients than a weaker noisier signal for transparent reconstruction (Figure 6). An additional weighting term measuring the efficiency of coding for each frame of input signal was thus proposed for budget allocation.

In the following, we introduce an improved budget allocation strategy. We also present the results of a perceptual quality study aimed at evaluating our novel technique and several possible importance metrics used for prioritizing source signals.

4.1 Improved budget allocation

The pinnacle value. Contrary to [Tsingos 2005], our improved budget allocation pre-computes the explicit number of STFT coefficients necessary to transparently reconstruct the original signal. This value, that we call *pinnacle*, is pre-computed for each time-frame of input audio data and stored together with the progressive

STFT representation. To compute the pinnacle value, we first sort the STFT coefficients by decreasing modulus order. The energy of each coefficient is integrated until a threshold of at least 99.5% of the total energy of the frame is reached and the number of corresponding coefficients is greater than $(1-tonality)N/2$, where N is the total number of complex Fourier coefficients in the frame and $tonality \in [0,1]$ is the tonality index of the frame [Painter and Spanias 2000; Kurniawati et al. 2002]. This index is close to 1 for tonal signals and drops to 0 for noisier signals.

Iterative importance sampling. We assume a constant number of arithmetic operations will be required for each complex STFT coefficient. Hence, fitting a budget number of operations for our pipeline at each processing frame directly amounts to selecting a budget number of coefficients for each frame of input sound signal. We can take advantage of pre-storing our FFT in decreasing energy order by directly processing the n_i first coefficients for each input signal s_i, so that the sum of all n_is does not exceed our total budget N. To determine the n_is, we first assign an importance value to each signal. This importance value can typically be the energy or loudness of the signal as proposed in [Tsingos et al. 2004; Tsingos 2005]. In this work, we also experimented with a *saliency* value derived from the model recently proposed in [Kayser et al. 2005]. This model is very similar to the visual saliency maps [Itti et al. 1998] but it is applied on a time-frequency domain representation of audio signals. In our case, after computing the auditory saliency map, we integrated saliency values over a small number of frequency subbands (we typically use 4 on a non-linear frequency scale).

Then, every input signal gets assigned a number of bins n_i relative to its relative importance as follows:

$$n_i = I_i / \sum_i I_i \cdot \text{targetCoeffs} \quad (1)$$

where I_i is the importance of the source i. Ideally, n_i should be smaller than the signal's pinnacle value to avoid suboptimal budget allocation as can be the case with the approach of [Tsingos 2005]. To avoid such situations, all remaining coefficients above pinnacle threshold are re-assigned to the remaining signals, that do not already satisfy the pinnacle value criterion. This allocation is again relative to the importance values of the signals:

$$n_i += I_i / \sum_i I_i \cdot \text{extraCoeffs} \quad (2)$$

The relative importance of each remaining signal is updated according to the reallocation of coefficients. If the budget is high enough, the process is iterated until all signals satisfy the pinnacle criteria or receive a maximal number of coefficients.

4.2 Quality evaluation study

To evaluate possible importance metrics and evaluate our pinnacle-based algorithm we conducted a quality evaluation study.

Experimental procedure. Seven subjects aged from 23 to 40 and reporting normal hearing volunteered for five different test sessions. For each session, we used a *Multiple Stimuli with Hidden Reference and Anchors* procedure (MUSHRA, ITU-R BS.1534) [Stoll and Kozamernik 2000; EBU 2003; International Telecom. Union 2001-2003]. Subjects were asked to simultaneously rank a total of 15 stimuli relative to a reference stimulus on a continuous 0 to 100 quality scale. The highest score corresponds to a signal indistinguishable from the reference. The reference stimuli were different mixtures of ambient, music and speech signals. In all cases, 12 of the 15 test-stimuli consisted of degraded versions of the mixture computed using our progressive mixing algorithm at various budgets (5%, 10% and 25% for music and ambient and 2%, 5% and 10% for speech), using our pinnacle-based technique, not using the pinnacle and using either loudness or saliency-based prioritization. In all cases, our processing is done using 32-bit floating point arithmetic and reconstructs signals at 44.1KHz. Two anchor stimuli, providing reference degradations, were also included. In our case, we chose a downsampled 16KHz/16-bit version of the mixture and a *mp3*-encoded version at 64Kbps. Finally, a hidden reference was also included. Stimuli were presented over headphones. Subjects could switch between stimuli at any point while listening. They could also define looping regions to concentrate on specific parts of the stimuli. A volume adjustment slider was provided so that subjects could select a comfortable listening level.

Results. Our study confirmed that the scalable processing approach is capable of generating high quality results using 25% of the original audio data and produces acceptable results with budgets as low as 10%. In the case of speech signals, for which the STFT representation is sparser, the algorithm could generate an acceptable mixture (avg. score 56/100) with only 2% of the original coefficients. As can be seen on Figure 7 (left), our approach yields significantly better results than a 16KHz reference signal (16KHz processing would correspond to a 30% reduction of data compared to our 44.1KHz processing). At 25% budget (or 10% in the case of speech), we obtain results comparable or better than the 64Kbps mp3-encoded reference. We performed an analysis of variance (ANOVA) [Howell 1992] on the results. As expected, the analysis confirmed a significant effect of the computing budget ($p<0.01$) on the quality of the resulting signal (Figure 7 right). We can see that the variation of perceived qualty is not generally a linear function of budget, especially for more tonal signals (music, speech) which can be efficiently encoded until a sharp breakdown point is reached. Interaction between budget and importance metric was found to be significant ($0.05<p<0.01$). At low or high budgets, the two metrics lead to very similar results. However, for intermediate budgets, loudness-based prioritization improved perceived quality relative to the saliency-based alternative. Similarly, using our new pinnacle algorithm also leads to a slight improvement in the results, especially for cases where both tonal and noisier signals are present. Noisier stationary sounds, which do not contain strong spectral features, usually receive a lower saliency value although they might contain significant energy and require more coefficients to be properly reconstructed. We believe this might explain why saliency-based prioritization led to lower perceived quality in some cases.

5 Cross-modal effects for sound scene simplification

In the preceding sections, we have improved different aspects of audio rendering for complex scenes, without consideration for the corresponding visuals. Intuitively, it would seem that such interaction of visual and audio rendering should be taken into account, and play a role in the choice of metrics used in the audio clustering algorithm. A first attempt was presented in [Tsingos et al. 2004], but was inconclusive presumably due to the difficulties with speech stimuli, which are generally considered to be a special case.

Research in ventriloquism (see Section 2), could imply that we should be more tolerant to localization errors for sound rendering when we have accompanying visuals. If this were the case, we could change the weighting terms in the clustering algorithm to create fewer clusters for sound sources in the visible frustum. However, a counter argument would be that in the presence of visuals,

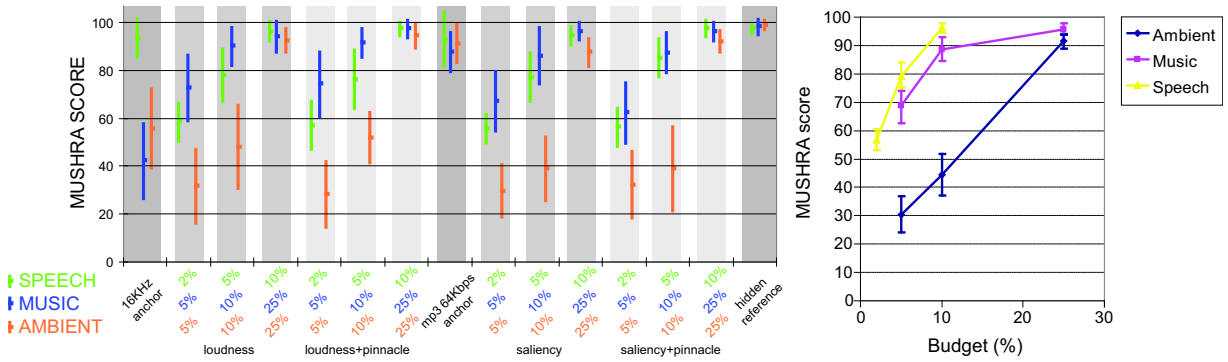

Figure 7: Left: Average MUSHRA scores and 95% confidence intervals for our progressive processing tests. Right: Average MUSHRA scores and 95% confidence intervals as a function of budget. Note how perceived quality does not vary linearly with the processing budget and also varies depending on the type (i.e., sparseness) of the sounds.

we are more sensitive to localization, and we should favour more clusters in the viewing frustum.

Our goal was to see whether we could provide some insight into this question with a pilot perceptual study. The next step was to develop and test an improved audio clustering algorithm based on the indications obtained experimentally.

5.1 Experimental setup and methodology

We chose the following experimental setup to provide some insight on whether we need more clusters in the visible frustum or not.

The subjects are presented with a scene composed of 10 animated - but not moving - objects emitting "ecologically valid" sounds, i.e., a moo-ing sound for the cow, a helicopter sound, etc. (Figure 8; also see and hear accompanying video).

We have two main conditions: audio only (i.e., no visuals) (condition *A*) and audio-visual (*AV*). Within each main condition we have a control condition, in which sources follow a uniform angular distribution, and the condition we test, where the proportion of clusters in the visible frustum and outside the visible frustum is varied.

We ran our test with 6 subjects (male, aged 23-45, with normal or corrected to normal vision, reporting normal hearing). All were naive about the experiment. Five of them had no experience in audio. Prior to the test, subjects were familiarized with isolated sound effects and their corresponding visual representation.

The subject stands 1 meter away from a 136 x 102 cm screen (Barco Baron Workbench), with an optical headtracking device (ART) and active stereo glasses (see the video). The field of view in this large screen experiment is approximately 70 °.

Headphones are used for audio output and our system uses binaural rendering [Blauert 1997; Møller 1992] using the LISTEN HRTF database (http://recherche.ircam.fr/equipes/salles/listen/). Our subjects were not part of the database. Hence, they performed a "point and click" pre-test to select the best HRTFs over a subset of 6 HRTF selected to be "most representative" similar to [Sarlat et al. 2006]. The marks attributed for the test are given with a joystick.

The *A* condition was presented first for three candidates, while *AV* condition was presented first for the other three. No significant effect of ordering was observed.

To achieve the desired effect, objects are placed in a circle around the observer; 5 are placed in the viewing frustum and 5 outside. For both control and main conditions, four configurations are used

Figure 8: An example view of the experimental setup for the audio-visual pilot user study.

randomly, by varying the proportion of clusters. Condition 1/4 has one cluster in the view frustum and 4 outside, 2/3, has 2 in the view frustum and 3 outside, etc. A uniform distribution of clusters corresponds to condition 1/4, with only 1 cluster in the frustum. Each condition is repeated 15 times with randomized object positions; these repetitions are randomized to avoid ordering effects.

We used the ITU-recommended *triple stimulus, double blind with hidden reference* technique [and 1993; ITU-R 1994]: 2 versions of the scene were presented ("A" and "B") and a given reference scene which corresponds to unclustered sound rendering. One of the 2 scenes was always the same as the reference (a *hidden reference*) and the other one corresponds to one of our clustering configurations. For each condition, the subject was presented with a screen with three rectangles ("A", "R" and "B"), shown in Fig. 8. The subjects were given a gamepad, and were instructed to switch between "A", "B" and "R" using three buttons on the pad, which were highlighted depending on the version being rendered. The subjects were asked to compare the quality of the approximations ("A" or "B") compared to the reference. They were asked to perform a "*quality judgment paying particular attention to the localization of sounds*" for the 2 test scenes, and instructed to attribute one of 4 levels of evaluation "No difference", "Slightly different", "Different" and "Clearly different" from the reference, which were indicated in rectangles next to the letter indicating the scene version (see Fig. 8 and accompanying video).

5.2 Analysis and results

We attributed a mark for each evaluation (from 0 to 3). As suggested by this ITU-R standard protocol, we only kept the difference between the test sample and the hidden reference. We also normal-

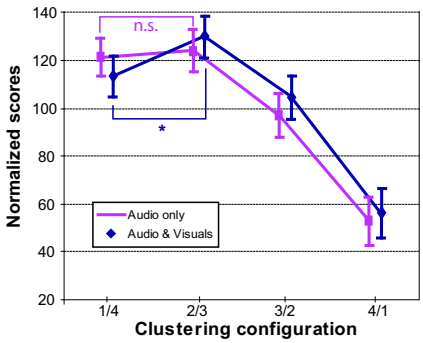

Figure 9: Mean values and 95% confidence intervals (N=6) in *A* and *AV* conditions as a function of the number of clusters inside/outside the view frustrum. For *AV*, the 2/3 configuration gives the best quality scores, which is not the case in the *A* condition. The "*" underlines that quality judgements in 1/4 and 2/3 cluster configurations for *AV* are significantly different ($p<0.05$), while the same comparison is non significant (n.s.) in the *A* condition.

ized the data by dividing each mark by the mean score of the user (the average of all marks of the candidate over all his tests).

There was no significant difference between the *A* and *AV* conditions regarding the respective scores of each cluster configuration. However, the difference of quality ratings between configurations was not similar in the two conditions. In condition *A*, 1/4 and 2/3 configurations lead to a similar quality evaluation (see Figure 9). In condition *AV*, the best quality is perceived in configuration 2/3. While 2/3 and 1/4 configurations are not perceived differently in condition *A* (Wilcoxon test, N=90, T=640.5, Z=0.21, p=0,83), the quality scores of 2/3 configuration are higher than those of 1/4 configuration in condition *AV* (Wilcoxon test, N=90, T=306.5, Z=2.56, p=0.01).

Overall, we consider the above results as a significant indication that, when we use the audio clustering algorithm with visual representation of the sound sources, it is better to have two clusters in the view frustrum, compared to a uniform angular distribution. This is indicated by the results for the 2/3 configuration, which is statistically different from all the other configurations in the AV condition. We expect this effect to be particularly true for scenes where there are visible sound sources in the periphery of the view frustum.

5.3 An audio-visual metric for clustering

Given the above observation, we developed a new weight in the clustering metric which encourages more clusters in the view frustum. We modify the cost-function of the clustering algorithm by adding the following weigthing term:

$$1+\alpha\left(\frac{\cos\theta_s-\cos\theta_f}{1-\cos\theta_f}\right)^n \qquad (3)$$

where θ_s is the angle between the view direction and the direction of the sound source relative to the observer, θ_f is the angular half-width of the view frustum and α controls the amplitude and n decay-rate of this visual improvement factor.

6 Implementation and Results

We ran tests on two scenes, one is a variant of the highway scene from [Tsingos et al. 2004], and another is a city scene. Both scenes

Figure 10: Left: the clusters without the audio-visual metric. Right: the clusters with our new metric. We clearly see that the new metric separates the sources appropriately.

are shown in Figure 1. In both cases, we used a platform with dual-core 3GHz *Xeon* processor and *NVidia 7950GX2* graphics accelerator; our system is built using the Ogre3D graphics engine and our custom audio library. Audio was processed at 44.1KHz using 1024-sample-long time-frames (i.e., 23 msec.). The following tests were performed with masking disabled to get a stable performance measure.

The highway scene contains 1004 sound sources, which are car engine and car stereo music sounds, cow "mooing" sound, train sounds, and water sounds in a stream. We found that a scalable premix budget of 25% is satisfactory in terms of audio quality (please hear and see the accompanying video). Comparing to the reference, we found that our entire perceptual processing pipeline resulted in an average signal-to-interference ratio of 18dB (min=5dB, max=30dB) for the sequence presented in the video. In this scene, clustering took 4.7 msec. per frame. Premixing was very simple and only included distance attenuation and accumulating source signals for each cluster. Premixing using 100% of the original audio data took 6 msec. Using our scalable processing with 25% budget we bring this cost down to 1.83 msec.

The street scene contains 1800 sound sources, which are footstep sounds and voices for the people in the crowd, car engine sounds, car radio sounds, bird sounds and sirens. Again, a scalable premix budget of 15% is satisfactory for this scene. Overall, we measured an average signal-to-interference ratio of 17dB (min=4dB, max=34dB) for the sequence presented in the video. In this scene, clustering took 5.46 msec. per frame while premixing using 100% of the original audio data took 6.84 msec. Using our scalable processing with 15% budget we bring the cost of premixing down to 2.17 msec.

In the commercial game *Test Drive Unlimited*, the average number of simultaneous sound sources is 130. A typical "player car" can generate up to 75 sound sources, while "AI" cars have a simplified model of maximum 42 sources. A maximum of 32 clusters were used in the game, although in a vast majority of cases 10 clusters are sufficient. Overall, the clustering approach discussed in Section 3.4 results in a reduction of 50-60% of CPU usage for the audio engine, which is freed for other application tasks, such as AI, gameplay etc.

To test the new audio-visual criterion, we constructed a variant of the street scene and an appropriate path, in which the positive effect of this criterion is clearly audible. For this test, we used $\alpha = 10$ and $n = 1.5$, which proved to be satisfactory. The scene used can be seen and heard in the video; the user follows a path in the scene (see accompanying video) and stops in a given location in the scene. We have 132 sources in the scene and target budget of 8 clusters. By switching between the reference, and the approximations with and without the audio-visual metric, we can clearly hear the improvement when more clusters are used in the view frustum. In particular, the car on the right has a siren whose sound is audibly displaced towards the centre with the audio-only metric.

7 Discussion and Conclusion

In this paper, we proposed a fast hierarchical clustering approach that can handle large numbers of sources and clusters. We also proposed a progressive processing pipeline for per-source effects (i.e., the premixing) that allows us to choose the best performance/quality ratio depending on the application and hardware constraints. Combined with auditory masking evaluation, these new algorithms allow for real-time rendering of thousands of mobile sound sources while finely controlling processing load vs. quality. In fact, while designing the test examples for the paper, the major problem we faced was authoring environments complex enough and most of the performance limitations actually came from the graphics engine. However, with next-generation games making increased used of procedurally-generated audio (e.g., based on physics engines), scenes with thousands of sound events to process are likely to become common in the near future. In our examples we only used simple per-source processing. However, in most gaming applications each source is likely to be processed with a chain of various effects (e.g., occlusion filters, echoes, re-timing, pitch-shifting, etc.) that would make our scalable approach even more attractive.

We also presented our perceptual studies for clustering and scalable premixing. A cross-modal perceptual study aimed at determining possible influence of the visuals on the required quality for audio clustering. Although one could expect ventriloquism to allow for rendering simplifications for visible sources, our study suggest that more clusters might actually be required in this case. A possible explanation for this is that, in a complex scene, clustering is likely to simplify auditory localization cues beyond common ventriloquism thresholds. As a consequence, we introduced a new metric to augment the importance of sources inside the view frustum. We demonstrated an example where, with a large number of sound sources outside the view frustum, it leads to improved results. We also performed a user-study of quality for the scalable premixing approach and showed that it leads to high quality results with budgets as low as 20 to 15% of the original input audio data. Although saliency-based importance appeared to show some limitations for our scalable processing algorithm compared to loudness, it might still be useful for prioritizing sources for clustering.

In the future, it would be interesting to experiment with auditory saliency metrics to drive clustering and evaluate our algorithms on various combinations of A/V displays (e.g., 5.1 surround or WFS setups). Also, the influence of ventriloquism on these algorithms merits further study. We also believe that authoring is now becoming a fundamental problem. Adapting our algorithms to handle combinations of sample-based and procedurally synthesized sounds seems a promising area of future research.

8 Acknowledgements

This research was funded by the EU IST FET Open project IST-014891-2 CROSSMOD (http://www.crossmod.org). We thank Autodesk for the donation of Maya, P. Richard and A. Olivier-Mangon for modelling/animation and G. Lemaitre for help with ANOVA.

References

ALAIS, D., AND BURR, D. 2004. The ventriloquism effect results from near-optimal bimodal integration. *Current Biology 14*, 257–262.

AND, C. G. 1993. Methods for quality assessment of low bit-rate audio codecs, proceedings of the 12th aes conference. 97–107.

BERKHOUT, A., DE VRIES, D., AND VOGEL, P. 1993. Acoustic control by wave field synthesis. *J. of the Acoustical Society of America 93*, 5 (may), 2764–2778.

BLAUERT, J. 1997. *Spatial Hearing : The Psychophysics of Human Sound Localization*. M.I.T. Press, Cambridge, MA.

CHEN, J., VEEN, B. V., AND HECOX, K. 1995. A spatial feature extraction and regularization model for the head-related transfer function. *J. of the Acoustical Society of America 97* (Jan.), 439–452.

DARLINGTON, D., DAUDET, L., AND SANDLER, M. 2002. Digital audio effects in the wavelet domain. In *Proceedings of COST-G6 Conference on Digital Audio Effects, DAFX2002, Hamburg, Germany*.

2003. EBU subjective listening tests on low-bitrate audio codecs. *Technical report 3296, European Broadcast Union (EBU), Projet Group B/AIM* (june).

FOUAD, H., HAHN, J., AND BALLAS, J. 1997. Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. *proceedings of the 1997 International Conference on Auditory Display (ICAD'97),*.

GALLO, E., LEMAITRE, G., AND TSINGOS, N. 2005. Prioritizing signals for selective real-time audio processing. In *Proc. of ICAD 2005*.

HAIRSTON, W., WALLACE, M., AMD B.E. STEIN, J. V., NORRIS, J., AND SCHIRILLO, J. 2003. Visual localization ability influences cross-modal bias. *J. Cogn. Neuroscience 15*, 20–29.

HERDER, J. 1999. Optimization of sound spatialization resource management through clustering. *The Journal of Three Dimensional Images, 3D-Forum Society 13*, 3 (Sept.), 59–65.

HOCHBAUM, D. S., AND SCHMOYS, D. B. 1985. A best possible heuristic for the ık-center problem. *Mathematics of Operations Research 10*, 2 (May), 180–184.

HOWELL, D. C. 1992. *Statistical methods for psychology*. PWS-Kent.

INTERNATIONAL TELECOM. UNION. 2001-2003. Method for the subjective assessment of intermediate quality level of coding systems. *Recommendation ITU-R BS.1534-1*.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (Nov.), 1254–1259.

ITU-R. 1994. Methods for subjective assessment of small impairments in audio systems including multichannel sound systems. itu-r bs 1116. Tech. rep.

JOT, J.-M., AND WALSH, M. 2006. Binaural simulation of complex acoustic scenes for interactive audio. *In 121th AES Convention, San Francisco, USA. Preprint 6950*.

JOT, J.-M., LARCHER, V., AND PERNAUX, J.-M. 1999. A comparative study of 3D audio encoding and rendering techniques. *Proceedings of the AES 16th international conference, Spatial sound reproduction, Rovaniemi, Finland* (April).

KAYSER, C., PETKOV, C., LIPPERT, M., AND LOGOTHETIS, N. 2005. Mechanisms for allocating auditory attention: An auditory saliency map. *Current Biology 15* (Nov.), 1943–1947.

KELLY, M., AND TEW, A. 2002. The continuity illusion in virtual auditory space. *proc. of the 112th AES Conv., Munich, Germany* (May).

KURNIAWATI, E., ABSAR, J., GEORGE, S., LAU, C. T., AND PREMKUMAR, B. 2002. The significance of tonality index and nonlinear psychoacoustics models for masking threshold estimation. In *Proceedings of the International Conference on Virtual, Synthetic and Entertainment Audio AES22*.

LANCIANI, C. A., AND SCHAFER, R. W. 1997. Psychoacoustically-based processing of MPEG-I layer 1-2 encoded signals. In *Proc. IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, 53–58.

LANCIANI, C. A., AND SCHAFER, R. W. 1999. Subband-domain filtering of MPEG audio signals. In *Proceedings of Intl. Conf. on Acoustics, Speech and Signal Processing*, 917–920.

LARCHER, V., JOT, J., GUYARD, G., AND WARUSFEL, O. 2000. Study and comparison of efficient methods for 3d audio spatialization based on linear decomposition of HRTF data. *Proc. 108th Audio Engineering Society Convention*.

LEWALD, J., EHRENSTEIN, W. H., AND GUSKI, R. 2001. Spatio-temporal constraints for auditory-visual integration. *Beh. Brain Research 121*, 1-2, 69–79.

MALHAM, D., AND MYATT, A. 1995. 3D sound spatialization using ambisonic techniques. *Computer Music Journal 19*, 4, 58–70.

MØLLER, H. 1992. Fundamentals of binaural technology. *Applied Acoustics 36*, 171–218.

PAINTER, E. M., AND SPANIAS, A. S. 2000. Perceptual coding of digital audio. *Proceedings of the IEEE 88*, 4 (Apr.).

SARLAT, L., WARUSFEL, O., AND VIAUD-DELMON, I. 2006. Ventriloquism aftereffects occur in the rear hemisphere. *Neuroscience Letters 404*, 324–329.

STOLL, G., AND KOZAMERNIK, F. 2000. EBU subjective listening tests on internet audio codecs. *EBU TECHNICAL REVIEW,* (June).

TOUIMI, A. B., EMERIT, M., AND PERNAUX, J.-M. 2004. Efficient method for multiple compressed audio streams spatialization. In *In Proceeding of ACM 3rd Intl. Conf. on Mobile and Ubiquitous multimedia*.

TOUIMI, A. B. 2000. A generic framework for filtering in subband domain. In *In Proc. of IEEE 9th Wkshp. on Digital Signal Processing, Hunt, Texas, USA*.

TSINGOS, N., GALLO, E., AND DRETTAKIS, G. 2004. Perceptual audio rendering of complex virtual environments. *Proc. SIGGRAPH'04* (August).

TSINGOS, N. 2005. Scalable perceptual mixing and filtering of audio signals using an augmented spectral representation. *Proc. of 8th Intl. Conf. on Digital Audio Effects (DAFX'05), Madrid, Spain* (Sept.).

WAND, M., AND STRASSER, W. 2004. Multi-resolution sound rendering. In *Symp. Point-Based Graphics*.

ZÖLZER, U., Ed. 2002. *DAFX - Digital Audio Effects*. Wiley.