Perceptual Audio Rendering of Complex Virtual Environments

Nicolas Tsingos, Emmanuel Gallo and George Drettakis
REVES/INRIA Sophia-Antipolis*

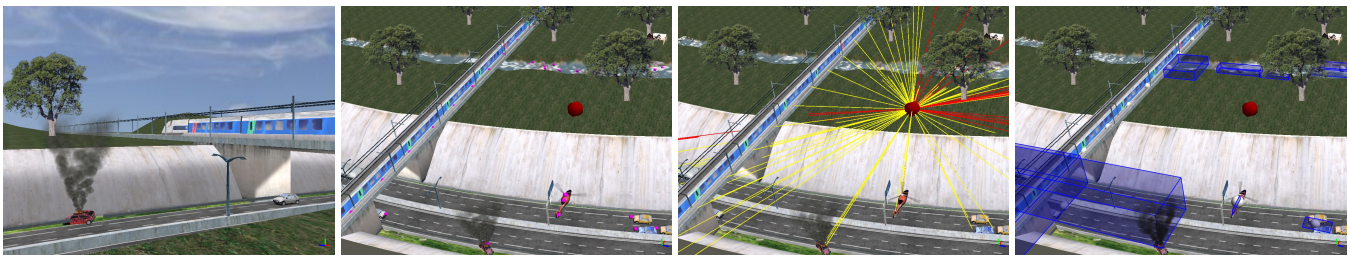

Figure 1: Left, an overview of a test virtual environment, containing 174 sound sources. All vehicles are moving. Mid-left, the magenta dots indicate the locations of the sound sources while the red sphere represents the listener. Notice that the train and the river are extended sources modeled by collections of point sources. Mid-right, ray-paths from the sources to the listener. Paths in red correspond to the perceptually masked sound sources. Right, the blue boxes are clusters of sound sources with the representatives of each cluster in grey. Combination of auditory culling and spatial clustering allows us to render such complex audio-visual scenes in real-time.

Abstract

We propose a real-time 3D audio rendering pipeline for complex virtual scenes containing hundreds of moving sound sources. The approach, based on auditory culling and spatial level-of-detail, can handle more than ten times the number of sources commonly available on consumer 3D audio hardware, with minimal decrease in audio quality. The method performs well for both indoor and outdoor environments. It leverages the limited capabilities of audio hardware for many applications, including interactive architectural acoustics simulations and automatic 3D voice management for video games.

Our approach dynamically eliminates inaudible sources and groups the remaining audible sources into a budget number of clusters. Each cluster is represented by one impostor sound source, positioned using perceptual criteria. Spatial audio processing is then performed only on the impostor sound sources rather than on every original source thus greatly reducing the computational cost.

A pilot validation study shows that degradation in audio quality, as well as localization impairment, are limited and do not seem to vary significantly with the cluster budget. We conclude that our real-time perceptual audio rendering pipeline can generate spatialized audio for complex auditory environments without introducing disturbing changes in the resulting perceived soundfield.

Keywords: Virtual Environments, Spatialized Sound, Spatial Hearing Models, Perceptual Rendering, Audio Hardware.

*contact Nicolas.Tsingos@sophia.inria.fr or visit http://www-sop.inria.fr/reves/

1 Introduction

Including spatialized audio is a key aspect in producing realistic virtual environments. Recent studies have shown that the combination of auditory and visual cues enhances the sense of immersion (e.g., [Larsson et al. 2002]). Unfortunately, high-quality spatialized audio rendering based on pre-recorded audio samples requires heavy signal processing, even for a small number of sound sources. Such processing typically includes rendering of source directivity patterns [Savioja et al. 1999], 3D positional audio [Begault 1994] and artificial reverberation [Gardner 1997; Savioja et al. 1999].

Despite advances in commodity audio hardware (e.g., [SoundBlaster 2004]), only a small number of processing channels (16 to 64) are usually available, corresponding to the number of sources that can be simultaneously rendered.

Although point-sources can be used to simulate direct and low-order indirect contributions interactively using geometric techniques [Funkhouser et al. 1999], a large number of secondary "images-sources" are required if further indirect contributions are to be added [Borish 1984]. In addition, many real-world sources such as a train (see Figure 1) are extended sound sources; one solution allowing their improved, if not correct, representation is to simulate them with a collection of point sources, as proposed in [Sensaura 2001]. This further increases the number of sources to render. This also applies to more specific effects, such as rendering of aerodynamic sounds [Dobashi et al. 2003], that also require processing collections of point sources.

For all the reasons presented above, current state-of-the-art solutions [Tsingos et al. 2001; Fouad et al. 2000; Wenzel et al. 2000; Savioja et al. 1999], still cannot provide high-quality audio renderings for complex virtual environments which respect the mandatory real-time constraints, since the number of sources required is not supported by hardware, and software processing would be overwhelming.

To address this shortcoming, we propose novel algorithms permitting high-quality spatial audio rendering for complex virtual environments, such as that shown in Figure 1. Our work is based on the observation that audio rendering operations (see Figure 2) are usually performed for every sound source while there is significant psycho-acoustic evidence that this might not be necessary due to limits in our auditory perception and localization accuracy [Moore 1997; Blauert 1983].

Similar to the occlusion culling and level of detail algorithms widely used in computer graphics [Funkhouser and Sequin 1993], we introduce a dynamic sorting and culling algorithm and a spatial clustering technique for 3D sound sources that allows for 1) significantly reducing the number of sources to render, 2) amortizing costly spatial audio processing over groups of sources and 3) leveraging current commodity audio hardware for complex auditory simulations. Contrary to prior work in audio rendering, we exploit *a priori* knowledge of the spectral characteristics of the input sound signals to optimize rendering. From this information, we interactively estimate the perceptual saliency of each sound source present in the environment. This saliency metric drives both our culling and clustering algorithms.

We have implemented a system combining these approaches. The results of our tests show that our solution can render highly dynamic audio-visual virtual environments comprising hundreds of point-sound sources. It adapts well to a variety of applications including simulation of extended sound sources and indoor acoustics simulation using image-sources to model sound reflections.

We also present the results of a pilot user study providing a first validation of our choices. In particular, it shows that our algorithms have little impact on the perceived audio quality and spatial audio localization cues when compared to reference renderings.

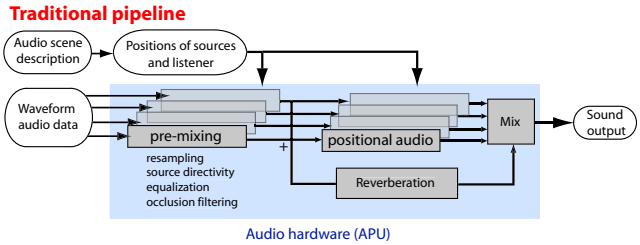

Figure 2: A traditional hardware-accelerated audio rendering pipeline. Pre-mixing can usually be implemented with few operations while positional audio and reverberation rendering require heavier processing.

2 Related Work

Our approach builds upon prior work in the fields of perceptual audio coding and audio rendering. The following sections give a short overview of the background most relevant to our problem.

Perceptual audio coding and sound masking

When a large number of sources are present in the environment, it is very unlikely that all will be audible due to masking occurring in the human auditory system [Moore 1997].

This masking mechanism has been successfully exploited in perceptual audio coding (PAC), such as the well known *MPEG I Layer 3 (mp3)* standard [Painter and Spanias 1997; Brandenburg 1999]. Note that contrary to PAC, our primary goal is to detect masking occurring between several sounds in a dense sound mixture rather than "intra-sound" masking. Since our scenes are highly dynamic, masking thresholds have to be continuously updated. This requires an efficient evaluation of the necessary information.

Interactive masking evaluation has also already been used for efficient modal synthesis [Lagrange and Marchand 2001; van den Doel et al. 2002; van den Doel et al. 2004] but, to our knowledge, no solution to date has been proposed to dynamically evaluate masking for mixtures of general digitally recorded sounds. Such techniques could nevertheless complement our approach for real-time synthesized sounds effects.

In the context of spatialized audio, binaural masking (*i.e.*, taking into account the signals reaching both ears) is of primary importance. Although *mp3* allows for joint-stereo coding, very few PAC approaches aim at encoding spatial audio and include the necessary binaural masking evaluation. This is quite a complex task since binaural masking thresholds are not entirely based on the spatial location of the sources but also depend on the relative phase of the signals at each ear [Moore 1997]. Finally, in the context of room acoustics simulation, several perceptual studies aimed at evaluating masking thresholds of individual reflections were conducted using simple image-sources simulations [Begault et al. 2001]. Unfortunately, no general purpose thresholds were derived from this work.

Spatial audio rendering

Few solutions to date have been proposed which reduce the overall cost of an audio rendering pipeline. Most of them specifically target the filtering operations involved in spatial audio rendering. Martens and Chen et al. [1987; 1995] proposed the use of principal component analysis of Head Related Transfer Functions (HRTFs) to speed up the signal processing operations. One approach, however, optimizes HRTF filtering by avoiding the processing of psycho-acoustically insignificant spectral components of the input signal [Filipanits 1994].

Fouad et al. [1997] propose a level-of-detail rendering approach for spatialized audio where the sound samples are progressively generated based on a perceptual metric in order to respect a budget computing time. When the budget processing time is reached, missing samples are interpolated from the calculated ones. Since full processing still has to be performed on a per source basis, the approach might result in significant degradation for large numbers of sources. Despite these advances, high-quality rendering of complex auditory scenes still requires dedicated multi-processor systems or distributed audio servers [Chen et al. 2002; Fouad et al. 2000].

An alternative to software rendering is to use additional resources such as high-end DSP systems (*Tucker Davis*, *Lake DSP*, etc.) or commodity audio hardware (e.g., *Sound Blaster* [SoundBlaster 2004]). The former are usually high audio fidelity systems but are not widely available and usually support ad-hoc APIs. The latter provide hardware support for game-oriented APIs (e.g., *Direct Sound 3D* [Direct Sound 3D 2004], and its extensions such as *EAX* [EAX 2004]). Contrary to high-end systems, they are widely available, inexpensive and tend to become *de facto* standards. Both classes of systems provide specialized 3D audio processing for a variety of listening setups and additional effects such as reverberation processing. In both cases, however, only a small number of sources (typically 16 to 64) can be rendered using hardware channels. Automatic adaptation to resources is available in *Direct Sound* but is based on distance-culling (far-away sources are simply not rendered) which can lead to discontinuities in the generated audio signal. Moreover, this solution would fail when many sources are close to the listener.

A solution to the problem of rendering many sources using limited software or hardware resources has been presented by Herder [1999a; 1999b] and is based on a clustering strategy. Similar approaches have also been proposed in computer graphics for off-line rendering of scenes with many lights [Paquette et al. 1998]. In Herder [1999a; 1999b], a potentially large number of point-sound sources can be down-sampled to a limited number of representatives which are then used as actual sources for rendering. In theory, such a framework is general and can accommodate primary sound sources and image-sources. Herder's clustering scheme is based on fixed clusters, corresponding to a non-uniform spatial subdivision, which cannot be easily adapted to fit a pre-defined budget. Hence, the algorithm cannot be used as is for resource management purposes. Second, the choice of the cluster representative (the

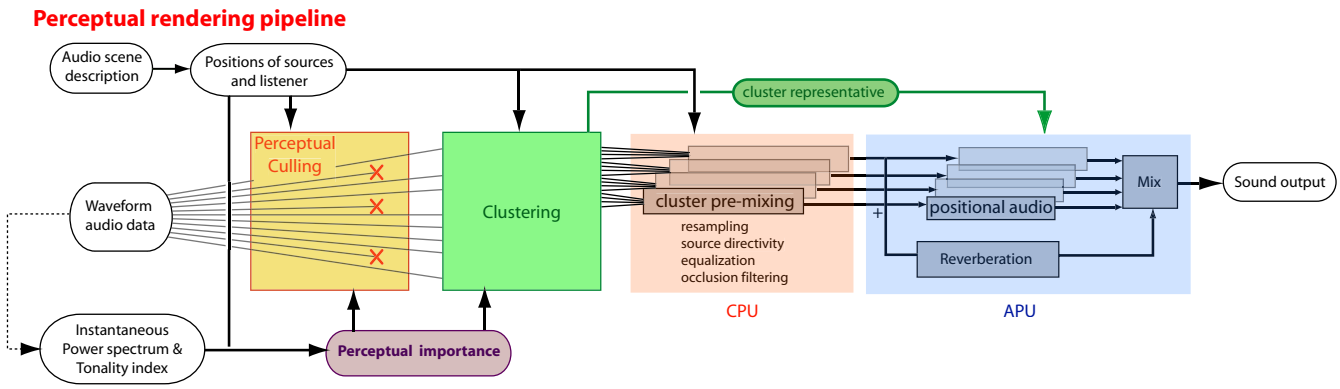

Figure 3: Our novel approach combining a perceptual culling/clustering strategy to reduce the number of sources and amortize costly operations over groups of sound sources.

Cartesian centroid of all sources in the cluster) is not optimal in the psycho-acoustical sense since it does not account for the characteristics of the input audio signals.

3 Overview of our contributions

We propose a novel spatial audio rendering pipeline for sampled sound signals. Our approach can be decomposed into four steps (see Figure 3) repeated for each audio processing frame through time (typically every 20 to 30 milliseconds):

- First, we evaluate the perceptual saliency of all sources in the scene. After sorting all sources based on their binaural *loudness*, we cull perceptually inaudible sources by progressively inserting sources into the mix until their combination masks all remaining ones. This stage requires the pre-computation of some spectral information for each input sound signal.

- We then group the remaining sound sources into a predefined budget number of clusters. We use a dynamic clustering algorithm based on the Hochbaum-Shmoys heuristic [Hochbaum and Schmoys 1985], taking into account the loudness of each source. A representative point source is constructed for each non-empty cluster.

- Then, an equivalent source signal is generated for each cluster in order to feed the available audio hardware channels. This phase involves a number of operations on the original audio data (filtering, re-sampling, mixing, etc.) which are different for each source.

- Finally, the pre-mixed signals for each cluster together with their representative point location can be used to feed audio rendering hardware through standard APIs (e.g., Direct Sound 3D), or can be rendered in software.

Sections 4 to 7 detail each of these steps. We have also conducted a pilot perceptual study with 20 listeners showing that our approach has very limited impact on audio quality and localization abilities. The results of this study are discussed in Section 8.

4 Perceptual saliency of sound sources

The first step of our algorithm aims at evaluating the perceptual saliency of every source. Saliency should reflect the perceptual importance of each source relative to the global soundscape.

Perception of multiple simultaneous sources is a complex problem which is actively studied in the community of auditory scene analysis (ASA) [Bregman 1990; Ellis 1992] where perceptual organization of the auditory world follows the principles of Gestalt psychology. However, computational ASA usually attempts to solve the inverse and more complex problem of segregating a complex sound mixture into discrete, perceptually relevant auditory events. This requires heavy processing in order to segment pitch, timbre and loudness patterns out of the original mixture and remains applicable only to very limited cases.

In our case, we chose the *binaural loudness* as a possible saliency metric. Due to sound masking occurring in our hearing process, some of the sounds in the environment might not be audible. Our saliency estimation accounts for this phenomenon by dynamically evaluating masked sound sources.

4.1 Pre-processing the audio data

In this paper, we focus on applications where the input audio samples are known in advance (*i.e.*, do not come from a real-time input and are not synthesized in real-time). Based on this assumption, we can pre-compute spectral features of our input signals throughout their duration and dynamically access them at runtime.

Specifically, for each input signal, we generate instantaneous short-time *power spectrum distribution (PSD)* and *tonality index* for a number of frequency sub-bands. Such features are widely used in perceptual audio coding [Painter and Spanias 1997].

The PSD measures the energy present in each frequency band, while the tonality index is an indication of the signal noisiness: low indices indicate a noisier component. This index will be used for interactive estimation of masking thresholds.

Our input sound signals were sampled at 44100 Hz. In order to retain efficiency, we use four frequency bands f corresponding to 0-500 Hz, 500-2000 Hz, 2000-8000 Hz and 8000-22050 Hz. Although this is far less than the 25 critical bands used in audio coding, we found it worked well in practice for our application while limiting computational overhead.

We derive our spectral cues from a short time fast Fourier transform (FFT) [Steiglitz 1996]. We used 1024 sample long Hanning-windowed frames with 50% overlap. We store for each band f its instantaneous power spectrum distribution (*i.e.*, the integral of the square of the modulus of the Fourier transform), $\mathbf{PSD}_t(f)$, for each frame t.

From the **PSD**, we estimate a log-scale *spectral flatness measure* of the signal as:

$$\mathbf{SFM}_t(f) = 10\ log10\left(\frac{\mu_g(\mathbf{PSD}_t(f))}{\mu_a(\mathbf{PSD}_t(f))}\right),$$

where μ_g and μ_a are respectively the geometric and arithmetic mean of the PSD over all FFT bins contained in band f. We then estimate the *tonality index*, $\mathbf{T}_t(f)$, as:

$$\mathbf{T}_t(f) = min(\frac{\mathbf{SFM}_t(f)}{-60}, 1).$$

Note that, as a result, $\mathbf{T}_t(f) \in [0,1]$.

This information is quite compact (8 floating-point values per frame, *i.e.*, 1.4 kbyte per second of input audio data at CD quality) and does not result in significant memory overhead.

This pre-processing can be done off-line or when the application is started but can also be performed in real-time for a small number of input signals since our unoptimized implementation runs about six times faster than real-time.

4.2 Binaural loudness estimation

At any given time-frame t of our audio rendering simulation, each source S_k is characterized by : 1) its distance to the listener r, 2) the corresponding propagation delay $\delta = r/c$, where c is the speed of sound, and 3) a frequency-dependent attenuation **A** which consists in a scalar factor for each frequency band. **A** is derived from the octave band attenuation values of the various filters used to alter the source signal, such as occlusion, scattering and directivity filters. For additional information on these filters see [Pierce 1984; ANSI 1978; Tsingos and Gascuel 1997; Savioja et al. 1999]. For instance, in the case of a direct, unoccluded contribution from the source to the receiver, **A** will simply be the attenuation in each frequency band due to atmospheric scattering effects. If the sound is further reflected or occluded, **A** will be obtained as the product of all scalar attenuation factors along the propagation path.

Our saliency estimation first computes the perceptual *loudness* at time t, of each sound source k, using an estimate of the sound pressure level in each frequency band. This estimate pressure level is computed at each ear as:

$$\mathbf{P}_t^k(f) = \mathbf{Spat}(S_k) \times \sqrt{\mathbf{PSD}_{t-\delta}^k(f)} \times \mathbf{A}_t^k(f)/r, \qquad (1)$$

where $\mathbf{Spat}(S_k)$ returns a direction and frequency dependent attenuation due to the spatial rendering (e.g., HRTF processing). In our case, we estimated this function using measurements of the output level of band-passed noise stimuli rendered with *Direct Sound 3D* on our audio board.

As a result, Equation 1 must be evaluated twice since the $\mathbf{Spat}(S_k)$ values will be different for the left and right ear.

The loudness values at both ears $\mathbf{Lleft}_t^k$ and $\mathbf{Lright}_t^k$, are then obtained from the sound pressure levels at each ear using the model of [Moore et al. 1997]. Loudness, expressed in *phons*, is a measure of the *subjective intensity* of a sound referenced to a 1kHz tone[1]. Based on Moore's model, we pre-compute a loudness table for each of our four frequency sub-bands assuming the original signal is a white noise. We use these tables to directly obtain a loudness value per frequency band given the value of $\mathbf{P}_t^k(f)$ at both ears.

Going back to linear scale, a scalar binaural loudness criterion L_t^k is computed as:

$$L_t^k = ||10^{\mathbf{Lleft}_t^k/20}||^2 + ||10^{\mathbf{Lright}_t^k/20}||^2. \qquad (2)$$

Finally, we normalize this estimate and average it over a number of audio frames to obtain smoothly varying values (we typically average over 0.1-0.3 sec. *i.e.*, 4-12 frames).

[1] by definition phons are equal to the sound pressure level, expressed in decibels, of a 1kHz sine wave.

4.3 Binaural masking and perceptual culling

We evaluate masking in a conservative manner by first sorting the sources by decreasing order according to their normalized loudness L_t^k and progressively inserting them into the current mix until they mask the remaining ones.

We start by computing the total power level of our scene

$$\mathbf{P_{TOT}} = \sum_k \mathbf{P}_t^k(f).$$

At each frame, we maintain the sum of the power of all sources to be added to the mix, $\mathbf{P_{toGo}}$, which is initially equal to $\mathbf{P_{TOT}}$.

We then progressively add sources to the mix, maintaining the current tonality $\mathbf{T_{mix}}$, masking threshold $\mathbf{M_{mix}}$, as well as the current power $\mathbf{P_{mix}}$ of the mix. We assume that sound power adds up which is a crude approximation but works reasonably well with real-world signals, which are typically noisy and uncorrelated.

To perform the perceptual culling, we apply the following algorithm, where **ATH** is the absolute threshold of hearing (corresponding to 2 phons) [Moore 1997]:

$\mathbf{M_{mix}} = -200$
$\mathbf{P_{mix}} = 0$
$\mathbf{T} = 0$
$\mathbf{P_{toGo}} = \mathbf{P_{TOT}}$
while $(dB(\mathbf{P_{toGo}}) > dB(\mathbf{P_{mix}}) - \mathbf{M}_{mix})$ **and** $(\mathbf{P_{toGo}} > \mathbf{ATH})$ **do**
  add source S_k to the mix
  $\mathbf{P_{toGo}} -= \mathbf{P}^k$
  $\mathbf{P_{mix}} += \mathbf{P}^k$
  $\mathbf{T} += \mathbf{P}^k * \mathbf{T}^k$
  $\mathbf{T_{mix}} = \mathbf{T}/\mathbf{P_{mix}}$
  $\mathbf{M_{mix}} = (14.5 + \mathbf{Bark}(\mathbf{f_{max}})) * \mathbf{T_{mix}} + 5.5 * (1 - \mathbf{T_{mix}})$
  k++
end

Similar to prior audio coding work [Painter and Spanias 1997], we estimate the masking threshold, $\mathbf{M_{mix}}(f)$ as:

$$\begin{aligned} \mathbf{M_{mix}}(f) &= (14.5 + \mathbf{Bark}(\mathbf{f_{max}})) * \mathbf{T_{mix}}(f) \\ &+ 5.5 * (1 - \mathbf{T_{mix}}(f)) \quad (dB), \end{aligned}$$

where $\mathbf{Bark}(\mathbf{f_{max}})$ is the value of the maximum frequency in each frequency-band f expressed in *Bark* scale. The Bark scale is a mapping of the frequencies in Hertz to Bark numbers, corresponding to the 25 critical bands of hearing [Zwicker and Fastl 1999]. In our case we have for our four bands: $\mathbf{Bark}(500) = 5, \mathbf{Bark}(2000) = 18, \mathbf{Bark}(8000) = 24, \mathbf{Bark}(22050) = 25$.

The masking threshold represents the limit below which a maskee is going to be masked by the considered signal.

To better account for binaural masking phenomena, we evaluate masking for left and right ears and assume the culling process is over when the remaining power at both ears is below the masking threshold of the current mix.

Since we always maintain an overall estimate for the power of the entire scene, our culling algorithm behaves well even in the case of a scene composed of many low-power sources. This is the case for instance with image-sources resulting from sound reflections. A naive algorithm might have culled all sources while their combination is actually audible.

5 Dynamic clustering of sound sources

Sources that have not been culled by the previous stage are then grouped by our dynamic clustering algorithm. Each cluster will act as a new point source representing all the sources it contains (*i.e.*, a point source with a complex impulse response). Our goal is to ensure minimal perceptible error between these auditory impostors and the original auditory scene.

5.1 Building clusters

Sources are grouped based on a distance metric. In our case, we use the sum of two spatial deviation terms from a source S_k to the cluster representative C_n: a distance deviation term and an angular deviation term:

$$d(C_n,S_k) = L_t^k \left(\beta log_{10}(||\mathbf{C_n}||/||\mathbf{S_k}||) + \gamma \frac{1}{2}(1 - \mathbf{C_n}.\mathbf{S_k}) \right), \quad (3)$$

where L_t^k is the loudness criterion calculated in the previous section (Eq. 2), $\mathbf{S_k}$ and $\mathbf{C_n}$ are the positions of source S_k and representative C_n expressed in a Cartesian coordinate system relative to the listener's position and orientation.

The weighting term L_t^k ensures that error is minimal for perceptually important sources. In our experiments we used $\beta = 2$ and $\gamma = 1$, to better balance distance and angle errors. Since human listeners perform poorly at estimating distances, our metric is non-uniform in distance space, resulting in bigger clusters for distant sources.

We use a dynamic clustering strategy based on the Hochbaum-Shmoys heuristic [Hochbaum and Schmoys 1985]. In a first pass, this approach selects n potential cluster representatives amongst all k sources by performing a farthest-first traversal of the point set using the metric of Eq. 3. In a second pass, sources are affected to the closest representative, resulting in a disjoint partitioning and clusters are formed. We also experimented with a global $k-means$ approach (e.g., [Likas et al. 2003]), with inferior results in terms of computing time. Both methods, however, gave similar results in terms of overall clustering error (the sum for every source of the distances as defined by Eq. 3).

The representative for the cluster must ensure minimal acoustic distortion when used to spatialize the signal. In particular it must preserve the overall impression of distance and incoming direction on the listener. Thus, a good starting candidate is the centroid of the set of points in (distance, direction) space. Since we are not using a fixed spatial subdivision structure as in [Herder 1999a], the Cartesian centroid would lead to incorrect results for spatially extended clusters. Using the centroid in polar coordinates yields a better representative since it preserves the average distance to the listener.

Moreover, source loudness will affect spatial perception of sound [Moore 1997]. Hence, we use our loudness criterion to shift the location of the representative once the clusters have been determined. The location of the representative is thus defined, in spherical coordinates relative to the listener's location, as:

$$\rho_{C_n} = \frac{\sum_j L_t^j r_j}{\sum_j L_t^j}, \;\; \theta_{C_n} = \theta(\sum_j L_t^j \mathbf{S}_j), \;\; \phi_{C_n} = \phi(\sum_j L_t^j \mathbf{S}_j), \quad (4)$$

where r_j is the distance from source S_j to the listener (S_j's are the sources contained in the cluster).

Figure 4 illustrates the results of our clustering technique in a simple outdoor environment and an indoor environment with sound reflections modeled as image-sources.

5.2 Spatial and temporal coherence

As a result of culling and loudness variations through time, our clustering process might produce different clusters from one frame to another. Since the clusters are mapped one-to-one with audio rendering buffers and the position of a cluster might switch abruptly, audible artefacts might be introduced. To avoid this problem, we perform a consistency check by comparing our current cluster distribution to the one obtained at the previous frame. We shuffle the order of our clusters so that the location of the $i-th$ cluster at frame t is as close as possible to the location of the $i-th$ cluster at frame $t-1$. We sort clusters using the summed loudness of all the sources they contain and perform the test greedily, by comparing distances between all pairs of clusters. Shuffling more perceptually relevant clusters first helps minimize possibly remaining artefacts.

6 Spatial rendering of clusters

The third stage of our pipeline is to compute an aggregate signal (or *pre-mix*) for an entire cluster based on the signals emitted by each individual sound source it contains. This signal will then be spatialized in the final stage of the pipeline.

6.1 Cluster pre-mixing

Computing this pre-mix of all sources involves a number of operations such as filtering (to account for scattering, occlusion, etc.), resampling (to account for the variable propagation delay and Doppler shifting) and $1/r$ distance attenuation of the input signals.

Filtering depends on the source directivity pattern or material properties in case of image-sources. Hence, it should be performed on each source individually. In our case, we use frequency dependent attenuation to account for all filtering effects. We implemented such filtering as a simple "equalization" over our four sub-bands.

For efficiency reasons, we pre-compute four band-passed copies of the original input signals. The filtered signal is then reconstructed as a sum of the band-passed copies weighted by the vector of attenuation values $\mathbf{A}$ (see Section 4.2).

Propagation delay also has to be accounted for on a per source basis. Otherwise, clicking artefacts appear as noticed in [Herder 1999a]. A simple method to account for time-varying non-integer delays is to re-sample the input sound signal. Simple linear interpolation gives good results in practice, especially if the signals are oversampled beforehand [Wenzel et al. 2000].

For maximum efficiency, we implemented these simple operations using SSE (Intel's Streaming SIMD Extensions) optimized assembly code.

Being able to pre-mix each source individually has several advantages. First, we can preserve the delay and attenuation of each source, ensuring a correct distribution of the energy reaching the listener through time. Doing so will preserve most of the spatial cues associated with the perceived size of the cluster and, more importantly, the timbre of the exact mix which would suffer from "comb-filter" effects if a single delay per cluster was used. This is particularly noticeable for reverberations rendered using image-sources. A second advantage of performing pre-mixing prior to audio hardware rendering is that we can provide additional effects currently not (or poorly) supported in existing audio hardware or APIs (e.g., arbitrary directivity patterns, time delays, etc.).

6.2 Spatializing clusters in hardware

The pre-mixed signals for each cluster, along with their representatives, can be used to auralize the audio scene in a standard spatialized audio system.

Each cluster is considered as a point-source located at the position of its representative. Any type of spatial sound reproduction strategy (amplitude panning, binaural processing, etc.) applicable to a point source model can thus be used.

Spatialization can be done in software, limiting the cost of spatial audio processing to the number of clusters. More interestingly, it can be done using standard "game-audio" APIs such as *Direct Sound (DS)*. In this case a hardware 3D audio buffer can be created for each cluster and fed with the pre-mixed signals. The sound buffer is then positioned at the location of the representative

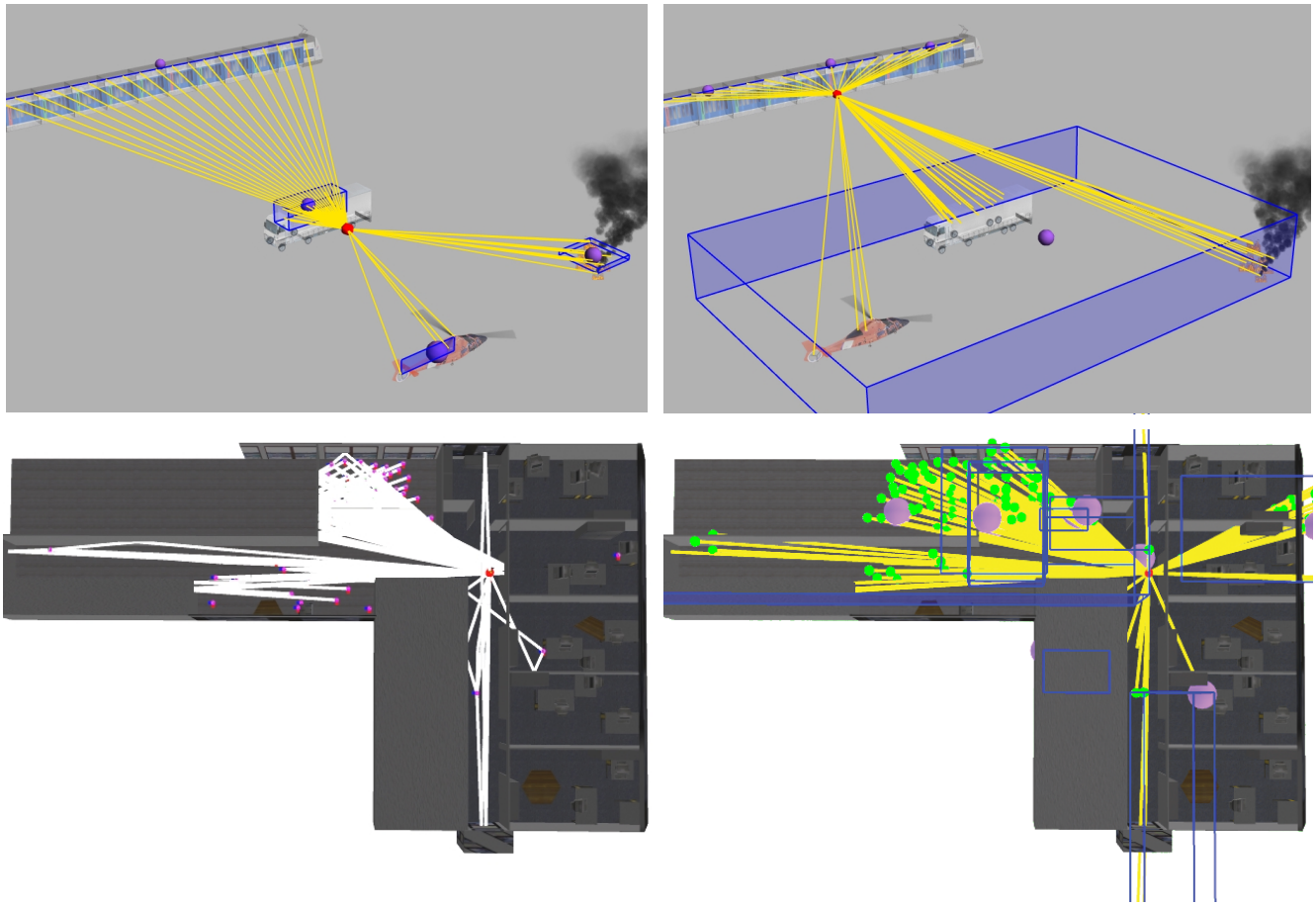

Figure 4: Top row: note how the four clusters (in blue) adapt to the listener's location (shown in red). Bottom row: a clustering example with image-sources in a simple building environment (seen in top view). The audible image-sources, shown in green in the right-hand side image, correspond to the set of reflection paths (shown in white) in the left-hand side image.

(e.g., using DS *SetPosition* command). We synchronize all positional commands at the beginning of each audio frame using DS *notification* mechanism. We also use a simple cross-fading scheme by computing a few extra samples at each frame and blending them with the first samples of the next prior to the spatialization. This eliminates artefacts resulting from sources moving in or out of clusters. In our current implementation, we use a 100-sample overlap at 44.1kHz (*i.e.*, 2ms or about a tenth of our audio frame).

Since audio hardware usually performs sampling-rate conversion, it is also possible to assign a different rate to each cluster depending on its importance. We define the importance of a cluster as the sum of the loudness values of all the sources it contains. We sort the clusters by decreasing importance prior to rendering, and map them to a set of hardware buffers whose sampling rate is decreasing, hence requiring less data to be rendered for an equivalent time-step. This is similar in spirit to the approach of [Fouad et al. 1997] but does not require extra software processing and better integrates with current hardware rendering pipelines.

Finally, we can also benefit from additional processing offered by current consumer audio hardware, such as artificial reverberation processing, as demonstrated in the video (see the trainstation and room acoustics sequences).

7 Applications and performance tests

We evaluated our algorithms on two prototype applications: 1) rendering of one exterior (*Highway*) and one interior (*Trainstation*) scene with numerous point sound sources and 2) rendering of an interior scene (*Busy office*) including modeling of sound reflections.

All tests were conducted on a *Pentium 4 3GHz* PC with a *nVidia GeForce FX5800 ultra* graphics accelerator and a *CreativeLabs Audigy2 platinum Ex* audio board. Audio was rendered using 1200-sample long audio frames at 44.1kHz.

Our first two examples feature *extended* sound sources resulting in many point sources to render. In our case, extended sources are collections of several point sources playing potentially different signals (e.g., the helicopter has 4 sound sources for rotors, jet exhaust and engine, the river is modeled with 40 point sources, etc.). Each sound source can have its own location, orientation and directivity function (e.g., the directivity of the jet exhaust of the helicopter and voice of the pedestrians in the train station are modeled using frequency dependent cosine lobes).

The train station example contains 60 people, with a sound source for their footsteps and one for their voices, two trains, with a source at each wheel, and a number of other sources (pigeons, etc.). A total of 195 sound sources are included. The highway scene contains 100 vehicles and environmental sound effects resulting in 174

environment	#sources	% culled	#clusters	loudness (ms)	culling (ms)	clustering (ms)	pre-mix (ms)	FPS w/o culling (Hz)	FPS w culling (Hz)
Trainstation	195	62	20	1.15	0.42	0.61	2.7	19	27
Highway	174	45	20	1.17	0.42	0.64	2.3	27	33
Busy office	355	71	20	3.8	0.8	1.14	2.5	< 1	22

Table 1: Computing time breakdown for three test environments and corresponding display frame rate (FPS) with and without culling.

Cluster range	All runs							Successful runs		
	localization time (s)			localization error (m)			found (%)	localization time (s)		
	avg.	min.	max.	avg.	min.	max.		avg.	min.	max.
1 to 4	90.40	8.32	408.45	0.49	0.00	4.12	85.2	66.08	8.32	272.86
5 to 7	52.46	7.74	151.31	0.52	0.00	3.16	88.5	34.16	7.74	113.08
8 to 10	66.60	9.18	270.64	0.21	0.00	1.00	100.0	51.98	9.18	127.20
11 to 13	52.43	6.76	204.51	0.32	0.00	2.24	82.1	49.55	6.76	204.51
14 to 16	72.11	8.19	320.82	0.27	0.00	1.00	100.0	61.42	8.19	298.76

Table 2: Statistics for localization time and error for all tests and successful tests only.

sources. The models used for visual rendering contain respectively about 70000 and 30000 polygons and no visibility optimization was used for display.

In the second type of applications we evaluated our rendering pipeline in the context of an interactive room acoustics simulation including the modeling of sound reflection of the walls of the environment. We used a path tracing algorithm to build the image-sources corresponding to sound reflections off the walls of a simple model (a small building floor containing 387 polygons). We simulated all direct paths and first-order reflections to the listener for 60 sound sources resulting in up to 360 image-sources to spatialize. Late reverberation was added to the simulation using the audio hardware's artificial reverberation engine. Note in the video how late reverberation varies as it is driven by the direct and first reflections resulting from our geometrical simulation.

Table 1 summarizes performance results for all sub-parts of our pipeline in the three environments. It also shows averaged video frame rate monitored with and without culling. Performing culling prior to clustering and pre-mixing is a key factor for the efficiency of the algorithm, since the number of sources to process is significantly reduced. In the *busy office* environment, rendering cannot run in real-time if culling is not applied.

Loudness, however, still has to be computed on a per-source basis so its calculation cost becomes significant as the number of sources increases. However, most of the calculation lies in the averaging process necessary to obtain smoothly varying values for the power and tonality estimates. It was our decision to leave this parameter as a variable but we believe averaging could be directly included in the pre-processing step of our approach.

8 Pilot validation study

In order to validate our perceptual rendering pipeline, we conducted a series of experiments aimed at evaluating the impact of culling and clustering sound sources on spatial audio rendering, both in terms of perceived audio quality and localization impairment. We also conducted cross-modal rendering tests to evaluate how visuals impact the perception of our spatial audio rendering.

8.1 Experimental conditions

We conducted our tests using non-individualized binaural presentation over headphones in a quiet office room. For spatial audio rendering, we used *Direct Sound 3D* accelerated by a *CreativeLabs Audigy2 platinum Ex* add-in board on a desktop computer. We used *Sennheiser HD600* headphones, calibrated to a reference listening level at eardrum (using a 1kHz sine tone of maximum amplitude).

The age of our 20 test subjects (12 males/8 females) ranged from 13 to 52 (averaging to 31.8). Most of them were computer scientists but very few having any experience in virtual reality or acoustics.

8.2 Clustering and localization impairment

Our first set of experiments aimed at evaluating the impact of clustering on the spatial perception of the virtual soundscape. In particular, we wanted to evaluate whether localization impairment arises from the clustering process in a task-based experiment. During this test, sound source masking was disabled.

Our experiment is similar in spirit to [Lokki et al. 2000] but uses a different experimental procedure. We asked the test subjects to perform a walkthrough in a 3D environment consisting of many small spheres located every meter on a 2D regular grid at listener's height. Sixteen sound sources corresponding to separate tracks (drums, vocals, guitars, etc.) of a short musical segment were randomly placed at some of these locations. The user was asked to locate an additional reference sound source, emitting a white noise signal, among all the spheres by pointing at it with the mouse and pressing the space bar. Only a single try was allowed. Navigation was done with the mouse, holding the buttons to go forward or backwards. The user could also rotate in place using the arrow-keys of the keyboard.

Each subject performed the test five successive times with a variable number of clusters ranging from small to large. The number of clusters was determined randomly for every test. All subjects underwent a short training period with a reference solution (without clustering) prior to performing the test to learn how to navigate and get accustomed to the 3D audio reproduction.

Subjects performed well in the localization task. Over the 100 runs of the experiment, the source was localized exactly 74% of the time and was found 90% of the time within a 1 meter range of its true location. These results are similar to the ones reported in [Lokki et al. 2000]. More than a half of our subjects localized the source with 100% accuracy. Table 2 reports localization time and error (distance of the selected sphere to the actual target sphere) for the five different cluster ranges. As can be seen, the number of clusters did not have a significant impact on localization time or accuracy.

Music				*Voice*				*Trainstation*									
								cluster rng	with graphics			w/o graphics			both		
cluster rng	avg.	min.	max.	cluster rng	avg.	min.	max.		avg.	min.	max.	avg.	min.	max.	avg.	min.	max.
1 to 4	1.625	-2	4	1 to 8	1.01	-1	3	1 to 8	0.37	0	2	0.35	-1	3	0.36	-1	3
5 to 7	0.433	-1	3	9 to 16	0.7	0	3	9 to 16	0.011	-1	1	0.25	0	2	0.189	-1	2
8 to 10	0.35	-1	3	17 to 24	0.875	0	2	17 to 24	0.364	-0.1	2	0.1	-1	1	0.281	-1	2
11 to 13	0.53	-1	3	25 to 32	0.6	-1	3	25 to 32	0.34	-0.1	2	0.5	0	2	0.42	-0.1	2
14 to 16	0.59	-1	3	33 to 40	0.65	0	3	33 to 40	1.1	-1	4	-0.05	-1	2	0.625	-1	4

Table 3: Statistics for *Reference minus Stimulus* marks for the *Music*, *Voice* and *Trainstation* environments (negative values correspond to cases where the hidden reference received a lower mark than the actual test stimulus).

8.3 Transparency of clustering and culling

The second set of experiments aimed at evaluating the transparency of the combined clustering and culling algorithms on the perceived sound quality.

We used the ITU-R[2] recommended *triple stimulus, double blind with hidden reference* technique, previously used for quality assessment of low bit-rate audio codecs [Grewin 1993]. Subjects were presented with three stimuli, R, A and B, corresponding to the reference, the test stimulus and a hidden reference stimulus[3]. The reference solution was a rendering with a single source per cluster and masking disabled.

Subjects could switch between the three stimuli at any time during the test by pressing the corresponding keys on the keyboard. The reproduction level could also be slightly adjusted around the calibrated listening level until the subject felt comfortable. Subjects were asked to rate differences between each test stimuli (A and B) and the reference R from "imperceptible" to "very annoying", using a scale from 5.0 to 1.0 (with one decimal) [ITU-R 1994].

We used two test environments, featuring different stimulus types. In the first environment (*Music*) the stimulus was a 16-track music signal where each track was rendered from a different location. The locations of the sources were randomized across tests. The second environment (*Voice*) featured a single source with a speech stimulus but also included the 39 first specular reflections from the walls of the environment (a simple shoebox-shaped room). The location of the primary sound source was also randomized across tests.

As before, each subject performed each test five successive times with a variable number of clusters ranging from small to large. The number of clusters was determined randomly for every test.

On average, 63% of the signals were masked during the simulation for the *Music* environment and 33% of the sources were masked in the *Voice* case. Table 3 reports detailed *Reference minus Stimulus* marks averaged over five cluster ranges. Our algorithm was rated 4.4 on average over all tests, very close to the reference according to our subjects (a mark of 4.0 corresponded to *"difference is perceptible but not annoying"* on our scale). This result confirms our hypothesis that our approach primarily trades spatial accuracy and number of sources for computing time while maintaining high restitution quality. For very low cluster budgets (typically 1 to 4) however, significant differences were reported, especially in the *Music* experiment. In such cases cluster locations can vary a lot from frame to frame in an attempt to best-fit the instruments with higher loudness values, resulting in an annoying sensation.

The room acoustics application, while being very well suited to our algorithms, is also very challenging since incorrectly culling image-sources might introduce noticeable changes in the level, timbre or perceived duration of the reverberation. Based on the results of our experiments, our algorithms were found to perform well at preserving the spatial and timbral characteristics of the reverberation in the *Voice* experiment. Our other room-acoustic test (*busy office*, shown in the video) confirms that our algorithm can automatically cull inaudible image-sources while largely preserving the auditory qualities of the reverberation.

Influence of visual rendering

We also attempted to evaluate the influence of possible interaction between image and sound rendering on the quality judgment of our perceptual audio rendering.

We repeated the above quality evaluation test using audio only and audio-visual presentation. Half of our subjects performed the test with visuals and half without. The test environment was a more complex train station environment featuring 120 sources (see video) and we had to limit our maximum number of clusters to 40 to maintain a good visual refresh rate.

Interestingly, the train station example received significantly better marks than the two other examples (see Table 3). This is probably due to its auditory complexity since it contains many simultaneous events, making it harder for the user to focus on a particular problem. For this test-case, the number of subjects was not high enough for a statistical validation of our results. Nonetheless, we note that the obtained marks are lower in the case where visuals were added. This is somewhat counter-intuitive since one could expect that the *ventriloquism effect*[4] would have helped the subjects compensate for any spatial audio rendering discrepancy [Vroomen and de Gelder 2004]. Actually, addition of graphics may have made it easier for the subjects to focus on discrepancies between the audio and visual rendering quality. In particular, some of our subjects specifically complained about having trouble associating voices with the pedestrians. We believe that our simple visual representation of the pedestrians (limited number of models, no facial animation) failed in this case to provide the necessary visual cues to achieve proper cross-modal integration of the voice and faces which is a situation we are highly sensitive to.

This indicates that a cross-modal importance metric should probably be used, possibly increasing the importance of visible sources (as suggested by [Fouad et al. 1997]) and that care should be taken in providing a sufficiently high degree of visual fidelity to avoid disturbing effects in cross-modal experiments.

9 Limitations of our approach

Based on this preliminary user study, our approach seems very promising although the tests were conducted using non-individualized binaural rendering. Using the test subjects' own measured HRTFs might reveal significant differences.

[2] International Telecommunication Union

[3] *i.e.*, the subjects did do not know which of A or B was the actual test or the reference.

[4] "presenting synchronous auditory and visual information in slightly separate locations creates the illusion that the location of the sound is shifted in the direction of the visual stimulus" [Vroomen and de Gelder 2004].

Although the method performs very well for a few hundred sound sources, it cannot easily scale to the thousands due to the cost of the clustering and pre-mixing algorithms. Loudness evaluation for every source would also become a significant bottleneck in this case. More efficient alternatives, such as a simple *A-weighting* of the pressure level could be used and should be evaluated. For such cases, synthesis algorithms might also be used to generate an equivalent signal for the cluster without having to pre-mix all the sources.

Our algorithm currently assumes input sound signals to be non-tonal (noise-like) and uncorrelated. However, the results of the preliminary study indicate that it does perform well on a variety of signals (music, voice, natural and mechanical sounds, etc.). Better results might be achieved by computing a finer estimate of the loudness by combining two values computed assuming the signal is closer to a noise or a tone in each frequency band. This would require determining a representative frequency for each frequency band during the pre-computing step (e.g., the spectral centroid) and using an additional pure-tone loudness table. Loudness values obtained under both assumptions could then be combined using the tonality index to yield a better loudness value. A finer estimate of the pressure level of the current mix and global scene would also improve the masking process. However, this is a more difficult problem for which we do not currently have a solution. Although we perform a sort of temporal averaging when estimating our criteria, we do not account for fine-grain temporal masking phenomena. This is a very interesting area for future research.

Our system currently uses pre-recorded input signals so that necessary spectral information can be pre-computed. However, we do not believe this is a strong limitation. Equivalent information could be extracted during the synthesis process if synthesized sounds are used. Our pre-processing step can also be performed interactively for a small number of real-time acquired signals (e.g., voice acquired from a microphone for telecommunication applications).

Finally, accuracy of the culling and clustering process certainly depends on the number of frequency bands used. Further evaluation is required to find an optimal choice of frequency bands.

10 Conclusion and future work

We presented an interactive auditory culling and spatial level-of-detail approach for 3D audio rendering of pre-recorded audio samples. Our approach allows for rendering of complex virtual auditory scenes comprising hundreds of moving sound sources on standard platforms using off-the-shelf tools. It leverages audio hardware capabilities in the context of interactive architectural acoustics/training simulations and can be used as an automatic 3D voice management scheme in video games. Our current pipeline can render more than ten times the number of sources that could be rendered using consumer 3D audio hardware alone. Hence, future audio APIs and hardware could benefit from including such a management scheme at a lower level (e.g., as part of the *DirectSound* drivers) so that it becomes fully transparent to the user.

We believe our techniques could also be used for dynamic coding and transmission of spatial audio content with flexible rendering, similar to [Faller and Baumgarte 2002]. This would be particularly useful for applications such as massively multi-player on-line games that wish to provide a spatialized "chat room" feature to their participants.

A pilot validation study shows that degradation in audio quality as well as localization impairment is very limited and does not seem to significantly vary with the number of used clusters.

We are currently preparing a full-blown follow-up study to provide additional statistical evaluation of the impact of our algorithm. Further experiments also need to be designed in order to evaluate several additional factors such as the pitch, beat or similarity of the signals in the culling and clustering process.

Our results so far suggest that spatial rendering of complex auditory environments can be heavily simplified without noticeable change in the resulting perceived soundfield. This is consistent with the fact that human listeners usually attend to one perceptual stream at a time, which stands out from the background formed by other streams [Moore 1997].

Acknowledgments

This research was supported in part by the *CREATE* 3-year RTD project funded by the 5th Framework Information Society Technologies (IST) Program of the European Union (IST-2001-34231), http://www.cs.ucl.ac.uk/create/. The authors would like to thank Alexandre Olivier for the modeling and texturing work. The train station environment in the video was designed, modeled and animated by Yannick Bachelart, Frank Quercioli, Paul Tumelaire, Florent Sacré and Jean-Yves Regnault. We thank Alias|***wavefront*** for the generous donation of their **Maya** software. Finally, the authors would like to thank Mel Slater for advice regarding the pilot validation study, Agata Opalach for thoroughly proof-reading the paper and the anonymous reviewers for their helpful comments.

References

ANSI. 1978. American national standard method for the calculation of the absorption of sound by the atmosphere. *ANSI S1.26-1978, American Institute of Physics (for Acoustical Society of America), New York*.

BEGAULT, D. R., MCCLAIN, B. U., AND ANDERSON, M. R. 2001. Early reflection thresholds for virtual sound sources. In *Proc. 2001 Int. Workshop on Spatial Media*.

BEGAULT, D. R. 1994. *3D Sound for Virtual Reality and Multimedia*. Academic Press Professional.

BLAUERT, J. 1983. *Spatial Hearing : The Psychophysics of Human Sound Localization*. M.I.T. Press, Cambridge, MA.

BORISH, J. 1984. Extension of the image model to arbitrary polyhedra. *J. of the Acoustical Society of America 75*, 6.

BRANDENBURG, K. 1999. mp3 and AAC explained. *AES 17th International Conference on High-Quality Audio Coding* (Sept.).

BREGMAN, A. 1990. *Auditory Scene Analysis, The perceptual organization of sound*. The MIT Press.

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.

CHEN, H., WALLACE, G., GUPTA, A., LI, K., FUNKHOUSER, T., AND COOK, P. 2002. Experiences with scalability of display walls. *Proceedings of the Immersive Projection Technology (IPT) Workshop* (Mar.).

DIRECT SOUND 3D, 2004. Direct X homepage, Microsoft©. http://www.microsoft.com/windows/directx/default.asp.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2003. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Transactions on Graphics 22*, 3 (Aug.), 732–740. (Proceedings of ACM SIGGRAPH 2003).

EAX, 2004. Environmental audio extensions 4.0, Creative©. http://www.soundblaster.com/eaudio.

ELLIS, D. 1992. A perceptual representation of audio. *Master's thesis, Massachusets Institute of Technology*.

FALLER, C., AND BAUMGARTE, F. 2002. Binaural cue coding applied to audio compression with flexible rendering. In *Proc. 113th AES Convention*.

FILIPANITS, F. 1994. Design and implementation of an auralization system with a spectrum-based temporal processing optimization. *Master thesis, Univ. of Miami*.

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), Xerox Palo Alto Research Center, Palo Alto, USA*.

FOUAD, H., BALLAS, J., AND BROCK, D. 2000. An extensible toolkit for creating virtual sonic environments. *Proceedings of Intl. Conf. on Auditory Display* (Atlanta, USA, May 2000).

FUNKHOUSER, T., AND SEQUIN, C. 1993. Adaptive display algorithms for interactive frame rates during visualization of complex virtual environments. *Computer Graphics (SIGGRAPH '93 proceedings), Los Angeles, CA* (August), 247–254.

FUNKHOUSER, T., MIN, P., AND CARLBOM, I. 1999. Real-time acoustic modeling for distributed virtual environments. *ACM Computer Graphics, SIGGRAPH'99 Proceedings* (Aug.), 365–374.

GARDNER, W. 1997. Reverberation algorithms. In *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds. Kluwer Academic Publishers, 85–131.

GREWIN, C. 1993. Methods for quality assessment of low bit-rate audio codecs. *proceedings of the 12th AES conference*, 97–107.

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.

HERDER, J. 1999. Visualization of a clustering algorithm of sound sources based on localization errors. *The Journal of Three Dimensional Images, 3D-Forum Society 13*, 3 (Sept.), 66–70.

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

ITU-R. 1994. Methods for subjective assessment of small impairments in audio systems including multichannel sound systems, ITU-R BS 1116.

LAGRANGE, M., AND MARCHAND, S. 2001. Real-time additive synthesis of sound by taking advantage of psychoacoustics. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, December 6-8*.

LARSSON, P., VÄSTFJÄLL, D., AND KLEINER, M. 2002. Better presence and performance in virtual environments by improved binaural sound rendering. *proceedings of the AES 22nd Intl. Conf. on virtual, synthetic and entertainment audio, Espoo, Finland* (June), 31–38.

LIKAS, A., VLASSIS, N., AND VERBEEK, J. 2003. The global *k*-means clustering algorithm. *Pattern Recognition 36*, 2, 451–461.

LOKKI, T., GRÖHN, M., SAVIOJA, L., AND TAKALA, T. 2000. A case study of auditory navigation in virtual acoustic environments. *Proceedings of Intl. Conf. on Auditory Display (ICAD2000)*.

MARTENS, W. 1987. Principal components analysis and resynthesis of spectral cues to perceived direction. In *Proc. Int. Computer Music Conf. (ICMC'87)*, 274–281.

MOORE, B. C. J., GLASBERG, B., AND BAER, T. 1997. A model for the prediction of thresholds, loudness and partial loudness. *J. of the Audio Engineering Society 45*, 4, 224–240. Software available at http://hearing.psychol.cam.ac.uk/Demos/demos.html.

MOORE, B. C. 1997. *An introduction to the psychology of hearing*. Academic Press, 4th edition.

PAINTER, E. M., AND SPANIAS, A. S. 1997. A review of algorithms for perceptual coding of digital audio signals. *DSP-97*.

PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 1998. A light hierarchy for fast rendering of scenes with many lights. *Proceedings of EUROGRAPHICS'98*.

PIERCE, A. 1984. *Acoustics. An introduction to its physical principles and applications*. 3rd edition, American Institute of Physics.

SAVIOJA, L., HUOPANIEMI, J., LOKKI, T., AND VÄÄNÄNEN, R. 1999. Creating interactive virtual acoustic environments. *J. of the Audio Engineering Society 47*, 9 (Sept.), 675–705.

SENSAURA, 2001. ZoomFX, MacroFX, Sensaura©. http://www.sensaura.co.uk.

SOUNDBLASTER, 2004. Creative Labs Soundblaster©. http://www.soundblaster.com.

STEIGLITZ, K. 1996. *A DSP Primer with applications to digital audio and computer music*. Addison Wesley.

TSINGOS, N., AND GASCUEL, J.-D. 1997. Soundtracks for computer animation: sound rendering in dynamic environments with occlusions. *Proceedings of Graphics Interface'97* (May), 9–16.

TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. *ACM Computer Graphics, SIGGRAPH'01 Proceedings* (Aug.), 545–552.

VAN DEN DOEL, K., PAI, D. K., ADAM, T., KORTCHMAR, L., AND PICHORA-FULLER, K. 2002. Measurements of perceptual quality of contact sound models. In *Proceedings of the International Conference on Auditory Display (ICAD 2002), Kyoto, Japan*, 345–349.

VAN DEN DOEL, K., KNOTT, D., AND PAI, D. K. 2004. Interactive simulation of complex audio-visual scenes. *Presence: Teleoperators and Virtual Environments 13*, 1.

VROOMEN, J., AND DE GELDER, B. 2004. Perceptual effects of cross-modal stimulation: Ventriloquism and the freezing phenomenon. In *Handbook of multisensory processes*, G. Calvert, C. Spence, and B. E. Stein, Eds. M.I.T. Press.

WENZEL, E., MILLER, J., AND ABEL, J. 2000. A software-based system for interactive spatial sound synthesis. *Proceeding of ICAD 2000, Atlanta, USA* (April).

ZWICKER, E., AND FASTL, H. 1999. *Psychoacoustics: Facts and Models*. Springer. Second Upadated Edition.