A Sound Spatialization Resource Management Framework

立体音響実現のための資源管理に関するフレームワーク

Jens Herder

Dissertation submitted to the
Institute of Information Sciences and Electronics,
University of Tsukuba

July 1999

Preface

Abstract

In a virtual reality environment, users are immersed in a scene with objects which might produce sound. The responsibility of a VR environment is to present these objects, but a practical system has only limited resources, including spatialization channels (mixels), MIDI/audio channels, and processing power. A sound spatialization resource manager, introduced in this thesis, controls sound resources and optimizes fidelity (presence) under given conditions, using a priority scheme based on psychoacoustics. Objects which are spatially close together can be coalesced by a novel clustering algorithm, which considers listener localization errors. Application programmers and VR scene designers are freed from the burden of assigning mixels and predicting sound source locations. The framework includes an abstract interface for sound spatialization backends, an API for the VR environments, and multimedia authoring tools.

ヴァーチャルリアリティ環境においてユーザは音源の存在する場面に注目するので、VR 環境はこれらの物体をユーザに正確に示さなければならない。しかし、現在のシステムでは空間化チャンネル (ミクセル)、MIDI/オーディオ チャンネル、処理能力を含む限られたリソースしか持っていない。この論文内で紹介される立体音響実現のための資源管理法は心理音響学による優先順位を参照することにより、与えられた状況下で音を管理し、より正確にする。空間内において互いに近くにある物体同士は、新しいクラスタリングアルゴリズム (人間の耳における情報収集の不確定さを考慮) により一つの物体とみなされる。アプリケーションプログラマや VR シーンの設計者はミクセルの割り当てや音源の場所の予想をする必要がなくなる。またそれは 3 次元音のバックエンドのための抽象的なインターフェイス、VR 環境向け API、マルチメディアオーサリングツールを含む。

Acknowledgments

The author is grateful for the encouragement received from his supervisor Nobuo Ohbo. The research was done at the University of Aizu within the Spatial Media Group. The University provides an excellent research environment, including an anechoic chamber for measuring filter functions and subjective listener tests. The colocated Multimedia Center has a sound spatialization system (i.e., the PSFC) for display to a medium-sized group. The author thanks Michael Cohen and William L. Martens for fruitful discussions and acknowledges the help of graduation research and master course students in implementing parts of some modules of the sound spatialization framework at the University of Aizu. Thanks also to Lothar M. Schmitt for his mathematics advice.

Dedicated to Miho and Bianca Mone

Jens Herder

Aizu-Wakamatsu, May 1999

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Sound spatialization is the processing of an audio stream in a virtual environment so that the audio stream is perceived from a specified location with source attributes like radiation pattern relative to listener location and ambiance. Included in this process is all the necessary audio rendering, like filtering by the media, reverberation and reflections. Sound spatialization resource management [Herder and Cohen, 1997] is the process of controlling a spatialization backend, which performs the actual audio rendering. Good resource management can be achieved only if the application-driven requirements are known and the audio rendering process is well understood. The context of sound spatialization resource management, the main theme of this thesis, is shown Figure 1.1. The upper layer includes multimedia or virtual reality applications which request audio rendering from an audio rendering system. This system has a well-defined sound spatialization application programmer interface and allows one to abstract the application from the resource management and vice versa. The resource management controls the audio rendering using the abstract sound spatialization backend interface.

Ease for application programmers Without a resource management system like that provided with the toolkit [Herder, 1998] developed by the author, application programmers must anticipate a lot of different configurations, with a consequent burden in programming spatial audio sources. During development of a systems which use spatial audio, the required spatialization resources and available spatialization resources are hard to predict (e.g., the number of participants in a chatspace application might vary, or the available spatialization resources on a certain computer system might be limited). Resource management eases the implementation by taking over the resource management processes, gradually scaling down the resource demand, depending on resource availability, in a perceptually optimal form.

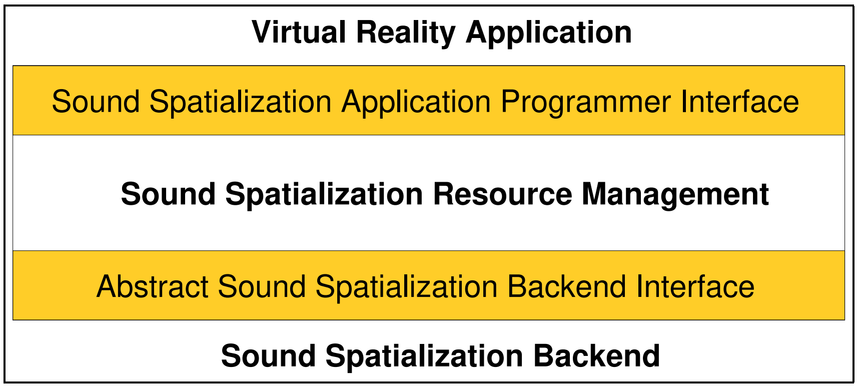


Figure 1.1: Resource management context

This has been also recognized by industry, which introduces resource management using priorities defined on the level of the application programmer interface. Early audio spatialization devices supported only a few of spatialization channels under the assumption that a listener could track only a small number of simultaneous moving sound events. User feedback and testing of applications show that application programmers have difficulties using sound spatialization channels effectively for best experience. Taking that into account, resource management, introduced on the level of the spatialization device and allowing more virtual channels, improves the spatial impression [Creative Technology Ltd., 1998].

1.1 Related research

Sound (source) prioritization for realtime scheduling can be performed by using a heuristic (estimate) for what a listener would attend [Fouad *et al.*, 1997]. This rating uses three factors: the listener's gaze (based on orientation response to an aural stimuli), the intensity of the sound (as a masking criteria), and the age of the sound (for modeling the adaptation response of the human aural system).

The number of calculated early reflections using an image source model can be dynamically adjusted to available processing power [Huopaniemi *et al.*, 1996]. As long as time is left within a processing cycle, higher-order reflections are calculated. Ranking is done from direct sound to higher orders, but not within a class (i.e., reflections of same order).

Common image sources (i.e., reflection) for several sound sources can be calculated in the case that the sound sources are close to each other [Savioja *et* *al.*, 1997, p. 44]. Such a representative image source is calculated by averaging multiple sound source into one sound source.

Predictors for human perception have been developed and applied in the field of global illumination. A perceptually-based visible difference predictor can be used as a criteria in an algorithm for adaptive mesh subdivision in image synthesis applications [Martens and Myszkowski, 1998].

A computational model for spatial hearing [Martin, 1995] might be used as a predictor for spatialization resource management. Such models have high computational costs.

Perceptual coding is used in different domains like audio (e.g., Dolby NR) and vision (e.g., MPEG and JPEG). Mainly masking effects are used to determine compressible information. In the auditory domain, masking is where a signal overwhelms another signal, which might be undesired noise, in a nearby frequency. The concepts in this thesis use also masking, but in the spatial domain. The main interest is to reduce the computational costs for spatialization and not reproduction (including transmission) of audio signals.

Sound spatialization is not surround sound (e.g., AC-3 Standard [Todd *et al.*, 1994]). In a surround sound system, listener and sound sources are usually at predetermined locations. In a sound spatialization system those can be freely placed depending on user or other data input. A surround sound system can be used to display auditory output from a sound spatialization system.

In general, localization task performance using only an auditory display is worse than using a visual display [Barfield *et al.*, 1997].

1.2 Applications of spatial sound

Spatial sound applications vary widely and are introduced in the following sections. The sections provide the reader with the context and helps to define requirements for a sound spatialization resource management framework.

1.2.1 Auditory feedback

Non-visual interfaces for visually disabled users employ spatial sound as a display and feedback device [Aritsuka and Hataoka, 1997]. Hierarchical graphical user interfaces can be extended by an auditory user interface applying auditory icons to convey user interface objects [Meinard, 1997], activated by moving a pointer (e.g., mouse-driven cursor) over certain objects. In the same way, an action applied to an object (e.g., pressing a button) activates an auditory icon giving information about the object and the applied action.

Such user interfaces address the needs of blind users but also benefit other users. Spatial sound can be used to convey the location of objects, including pointing indicators. Audio messages (earcons) [Blattner *et al.*, 1992, p. 89] can be used to report events to the user without disturbing a visual task like reading a text. An earcon coupled with spatial location can convey spatial location of an event or tighten the association of an earcon and a graphical object.

Acoustic cues can enhance the level of immersion provided by haptic applications [Ruspini and Khatib, 1998]. The objects in contact produce a sound depending on their materials and performed action (e.g., pushing, scratching, ...). A haptic interface provides sensation at a specific location. This sensation should be coupled with sound at same location for greater realism. Those sounds must be synchronized and controlled with the haptic events.

1.2.2 Teleconferencing

A sound spatialization system enhances teleconferencing applications [Aoki *et al.*, 1994]. Three primary benefits [Barfield *et al.*, 1997] of auditory spatial information displays are identified as:

1. Relieving processing demands on the visual modality,
2. Enhancing spatial awareness, and
3. Directing attention to important spatial events.

Speech intelligibility degrades as target and distracting source become closer [Hawley *et al.*, 1996]. On the other hand, judgment of the number of concurrent voices does not improve with spatialization [Kashino and Hirahara, 1996].

The well-known "cocktail party effect" [Arons, 1992] can be used for enhancing speech recognition. Talker identification can be eased by matching visual and acoustical location or each talker has an own unique acoustical spatial location. Auditory cues like talking at close range can suggest communication modalities like `confide` as described in the following sections.

A Chatspace is a VR environment which enables people to meet and communicate in cyberspace. People are represented as avatars, graphical objects (Figure 1.2) with behavior controlled by each participant. Such an environment will be dramatically improved using voice spatialization.

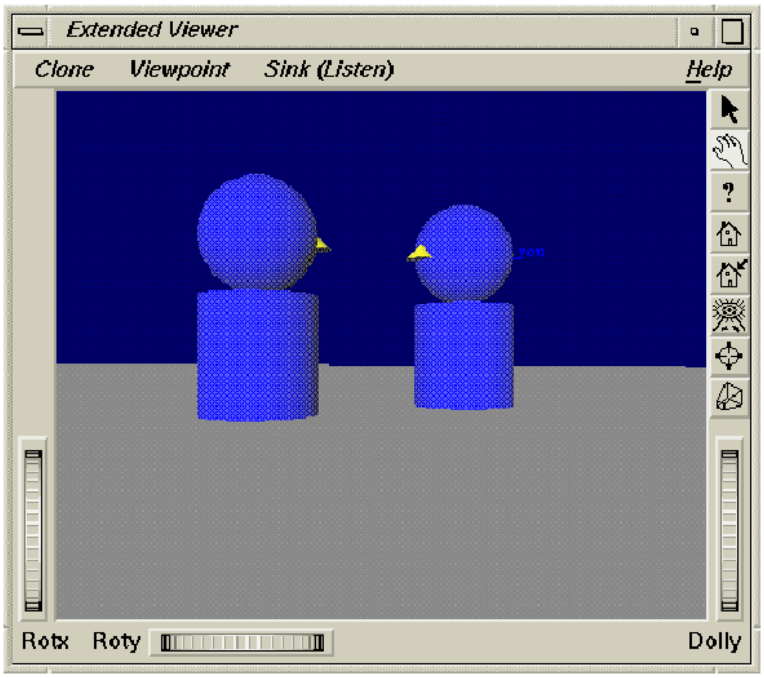


Figure 1.2: Chatspace: people meet and communicate in cyberspace

Symbolic representations of exclude and include for sound sources and sinks

Figurative suggestions of `mute/solo` & `cue` and `deafen/confide` & `harken` are described for exclude and include for sound sources and sinks [Cohen and Herder, 1998]. Those selection functions for the communication channels have direct influence on the resource management. On a logical level, sound source and sinks (receivers, generalization of listener and microphones) are resources assigned to users.

Shared virtual environments, e.g., chatspaces, require generalized control of user-dependent media streams. Traditional audio mixing idioms for enabling and disabling various sources employ `mute` and `solo` functions, which, along with `cue`, selectively disable or focus on respective channels. Exocentric interfaces which explicitly model not only spatial audio sources, but also location, orientation, directivity, and multiplicity of sinks, motivate the generalization of `mute/solo` & `cue` to exclude and include, manifested for sinks as `deafen/confide` & `harken`, a narrowing of stimuli by explicitly blocking out and/or concentrating on selected entities. This section introduces figurative representations of these functions, virtual hands to be clasped over avatars' ears and mouths, with orientation suggesting the nature of the blocking.

Virtual Mixing Non-immersive perspectives in virtual environments enable fluid paradigms of perception, especially in the context of frames-of-reference for conferencing and musical audition [Cohen, 1995] [Cohen, 1998]. Traditional mixing idioms for enabling and disabling various sources employ `mute` and `solo` functions, which, along with `cue`, selectively disable or focus on respective channels. Exocentric interfaces which explicitly model not only spatial audio sources, but also location, orientation, directivity, and multiplicity of sinks, described by Table 1.1, motivate the generalization of `mute/solo` & `cue` commands to exclude and include, manifested for sinks as `deafen/confide` & `harken`, a narrowing of stimuli by explicitly blocking out and/or concentrating on selected entities [Cohen, 1997] [Cohen and Koizumi, 1998]. (`harken` is used to describe focusing on one's own sink.)

	Role	
	Source	Sink
Function	radiation	reception
Level	amplification	sensitivity
Direction	output	input
Instance	speaker (human or loud-)	listener (human or dummy-head)
Organ	mouth	ear

Table 1.1: ${}^{\mathrm{s}}\mathrm{OU}^{\mathrm{rce}}_{\mathrm{Tput}}$ and ${}^{\mathrm{s}}\mathrm{IN}^{\mathrm{k}}_{\mathrm{put}}$

Exclude and Include Audio Functions A source can be disabled with `mute`; its complement `solo` disables all non-`solo`ed sources. The semantics of `mute` and `solo` can be described in predicate calculus notation:

$$\mathrm{active}(\mathrm{source}_{\mathrm{x}}) = \neg\mathrm{mute}(\mathrm{source}_{\mathrm{x}}) \ \wedge\ (\exists\ \mathrm{y}\ \mathrm{solo}(\mathrm{source}_{\mathrm{y}}) \ \Rightarrow\ \mathrm{solo}(\mathrm{source}_{\mathrm{x}})) \tag{1.1}$$

As sinks are duals of sources, the semantics of `deafen` and `confide` (& `harken`) are analogous:

$$\mathrm{active}(\mathrm{sink}_{\mathrm{x}}) = \neg\mathrm{deafen}(\mathrm{sink}_{\mathrm{x}}) \ \wedge\ (\exists\ \mathrm{y}\ \mathrm{confide}(\mathrm{sink}_{\mathrm{y}}) \ \Rightarrow\ \mathrm{confide}(\mathrm{sink}_{\mathrm{x}})) \tag{1.2}$$

These two predicates can be described by a generalized representation, using "exclude" to stand for `mute` and `deafen` and "include" to stand for `solo` and `confide` (& `harken`):

$$\text{active}(x) = \neg\text{exclude}(x) \;\wedge\; (\exists\, y\ \text{include}(y) \;\Rightarrow\; \text{include}(x)) \tag{1.3}$$

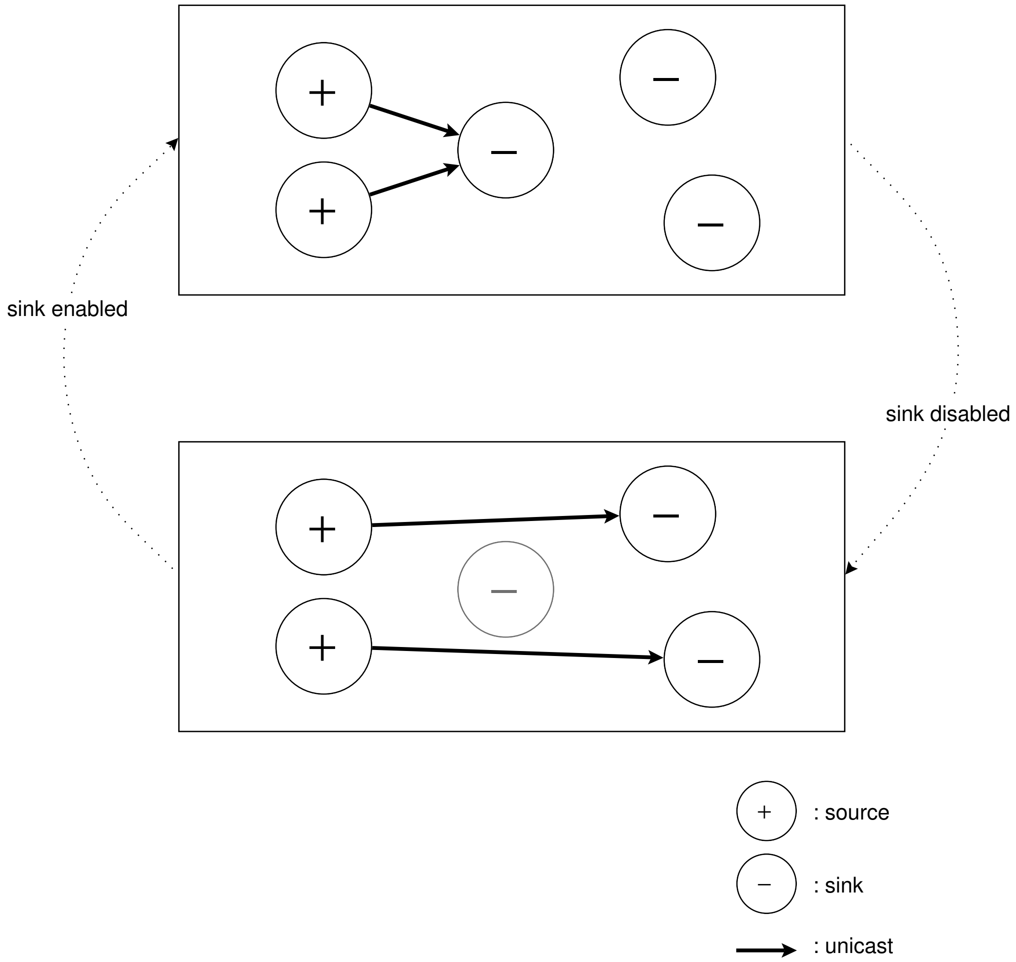


Figure 1.3: Unicast source → sink transmissions: orphaned sources adopted by sinks

Such functions can be applied not only to other users' sinks for privacy, but also to one's own sinks for selective presence [Herder and Cohen, 1996b]. Multiple sinks are useful in both groupware, where a common environment implies social inhibitions to rearranging shared sources like musical voices or conferees, and individual sessions in which spatial arrangement of sources, like the configuration of a concert orchestra, has mnemonic value. As shown by Figure 1.3, an "autofocus" mode can be used to adjust source→sink mappings, depending on sink activation.

User (human pilot)	Delegate (representative, projected presence)
human body	avatar
carbon community	electronic community
RL (**r**eal **l**ife)	virtual life
	synthespian (**synth**etic **thespian**)
motion capture & human actor	vactor (**v**irtual **actor**)

Table 1.2: User and delegate

Figurative Avatars

Representation of Exclude Audio Functions A human user can be represented in virtual space by one or more avatars, as suggested by Table 1.2. A figurative avatar in virtual space is naturally humanoid, including especially a head, since it not only embodies a center of consciousness, but also important communication organs: ears, mouth, and eyes. Exclude and include source and sink operations can be visually represented by iconic attributes which can distinguish between operations reflexive, invoked by a user associated with a respective icon, and transitive, invoked by another user in the shared environment. Distributed users might typically share spatial aspects of a groupware environment, with attributes like `mute`dness or `deafen`dness determined and displayed on a per-user basis.

For example, as shown in Table 1.3, a source representing a human teleconferee denotes `mute`edness with an iconic hand clasped over its mouth, oriented differently (thumb up or down) depending on whether the source was `mute`d by its owner (or one of its owners) or another, unassociated user. (In the former case, all the users in the space would observe the `mute`, but in the latter, only users disabling the remote source would typically see the `mute`.) An audio muffler could be wrapped around an iconic head to denote its deafness, but to distinguish between self-imposed deafness, invoked by a user whose attention is directed elsewhere, and distally imposed, invoked by a user desiring privacy, hands clasped over the ears should be oriented differently depending on the agent of deafness. As such attributes are orthogonal, simultaneously applied filters could be represented by interpenetrated virtual models.

Representation of Include Audio Functions Include functions (`solo` and `confide`) manifest visually as the respective complementary exclude functions applied to the complement of the appropriate selection. `solo` is implemented as a straight-forward extension of `mute`, effectively muting the

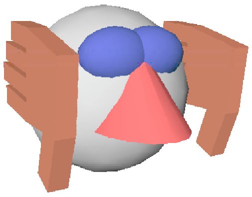
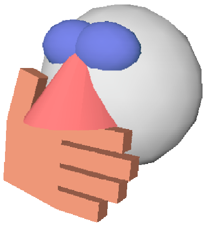
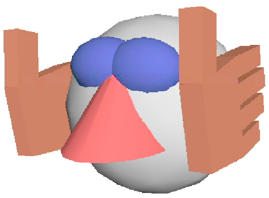
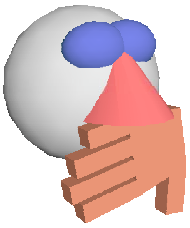
	action	
	deafen (muffle)	`mute` (muzzle)
object	sink	source
own		
	 (thumbs down)	 (thumb up)
other	  (thumbs up)	  (thumb down)

Table 1.3: Deafening and muting one's own and others' avatars

complement of the `solo`ed selection, as `confide` (& `harken`) `deafen`s the complement of the selection. These pairs of fields are maintained separately, however, anticipating a visual idiom that distinguishes the explicit and implicit operations. For example, a visual spotlight could be used to denote `solo`ed sources or `confide`d sinks.

avatar	own	other
mode	reflexive	transitive
sink	`deafen` `harken`	`deafen` `confide`
source	`mute` `solo`	`mute` `solo`

Table 1.4: Reflexive and transitive audio exclude and include operations

Groupware Actions along the main diagonal of Tables 1.3 and 1.4, i.e., applied to one's own sinks or to others' sources, manifest locally (in the respective user's spatialization process and soundscape), while actions along the secondary diagonal, i.e., applied to others' sinks or to one's own sources, manifest remotely (in other users' spatialization processes and soundscapes). Such medial attributes do not propagate to distal users' soundscapes; a normal user can `mute` a personally undesired source, but can't prevent its disturbing others.

1.2.3 Sonification

Sonification, auditory visualization, is the computer-based transformation of numeric data into human-interpretable acoustical information [Astheimer, 1995, p. 16–17].

Displays are used to monitor systems. Auditory displays can substitute for or extend graphical displays. A human listener is quite sensitive to changes in audio streams, which makes auditory displays suitable for monitoring tasks over a long period as long as the normal state does not require too much concentration. Monitoring complex processes necessitates multi-dimensional displays. An auditory system for sonification [Kramer, 1994, p. 187] can add several dimensions. Compared to a monoral audio system a sound spatialization system can provide at least three additional dimensions.

Visualization systems using data flow paradigm for modeling and user interface are extended by auditory interfaces for providing sonification [Astheimer, 1995, p. 103–126]. Such systems are easy to handle and inherit all benefits of state of the art visualization systems.

1.2.4 Spatial music — the Helical Keyboard project

Inspired by the cyclical nature of octaves and helical structure of a scale, a model of a piano-style keyboard (see Figure 1.4) was prepared for the helical keyboard project [Herder and Cohen, 1996a] [Herder and Cohen, 1999], which was then geometrically warped into a helicoidal configuration, one octave/revolution, pitch mapped to height. It can be driven by MIDI events, realtime or sequenced, which stream is both synthesized and spatialized by a spatial sound backend. The sound of the respective notes is spatialized with respect to sinks, avatars of the human user, by default in the tube of the helix.

The helical keyboard system is designed to allow, for instance, separate audition of harmony and melody, commonly played by the left and right hands, respectively, on a normal keyboard. Perhaps the most exotic feature of the helical keyboard is the ability to fork one's presence, replicating subject instead of object by installing multiple sinks at arbitrary places around a virtual scene so that, for example, harmony and melody can be separately spatialized, using two heads to normalize the octave; such a technique effectively doubles the helix from the perspective of a single listener. Rather than a symmetric arrangement of the individual helices, we perceptually superimpose them in-phase, coextensively, so that corresponding notes in different octaves are at the same azimuth.

The inherently limited number of sound spatialization resources motivated development of a spatialization resource manager, described in this thesis. To meet the need for better perceiving the large space of harmony and melody, the unique feature of multiple sinks in Section 2.2.3 was introduced. The Helical Keyboard project explores multiple acoustic presence, introduced in [Cohen, 1995] [Cohen and Koizumi, 1995], for three-dimensional space, including manipulations and different visualizations.

1.2.5 Virtual acoustics

Virtual acoustics can be divided into several fields depending on scale and application. In (virtual) room acoustics [Dalenbäck *et al.*, 1996] or architectural acoustics, the goal is to simulate the acoustic of rooms for improving or designing rooms, mainly concert halls. In this thesis, such simulations are

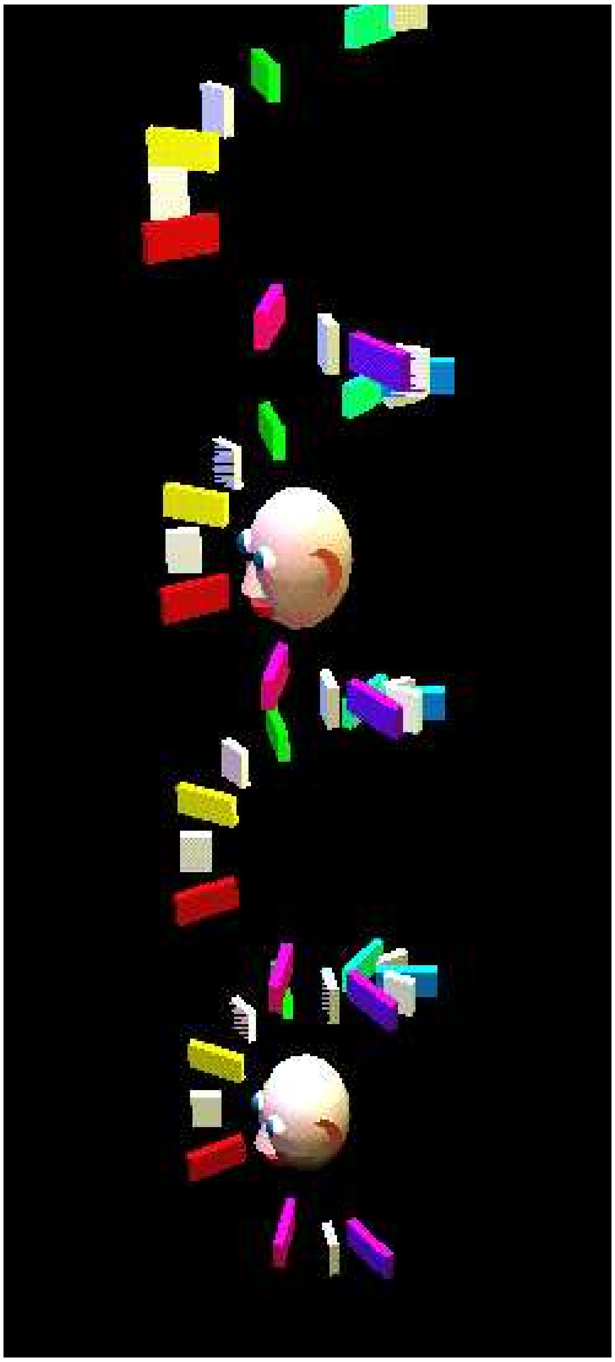


Figure 1.4: Helical Keyboard: an application

handled only briefly, because the involved complexity does not allow real-time applications on currently available systems without neglecting important acoustic properties. Virtual acoustics includes modeling of sound sources at the level of musical instruments like shown in virtual concerts [Cohen and Koizumi, 1995] where sound sources can be arranged. Virtual acoustics goes even further by modeling the physical properties of musical instruments and radiation patterns. Modeling of radiation patterns can be done physically using a set of arranged and adjusted loudspeakers [Caussé *et al.*, 1992] or virtually using source radiation transfer functions [Cook and Trueman, 1998]. It is also reasonable to split the modeling of an instrument into several sound sources (called "elementary sources" in [Karjalainen *et al.*, 1995]), each with its own radiation patterns. The sound of a musical instrument (e.g., clarinet) interacts with the environment (e.g., floor) and depends on the orientation of the instrument (as in Figure 1.5). The overall radiation pattern of a clarinet depends on which tone holes are open.

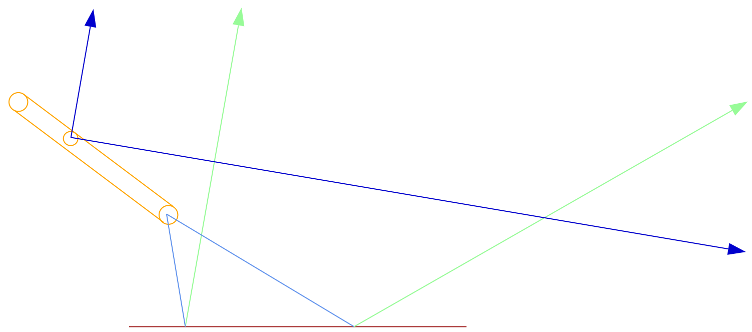


Figure 1.5: Multiple sources can be used to model complex radiation pattern

Physical modeling for sound synthesis is resource demanding. Managing and controlling sound synthesis can be achieved by monitoring resource availability and in case of over load requiring the involved synthesis modules to reduce demand by reducing frequency resolution [Goudeseune and Hamman, 1998].

1.3 Resource management requirements

Sound spatialization quality requirements for the resource management vary depending on the application. This means the resource manager needs to be configurable or adaptable for different applications. The same flexibility is required for the spatialization module (hard or software).

A teleconferencing application might place most sound sources into the horizontal plane. Applications for blind users would give higher priorities to sound sources which are involved with a user. Elementary sound sources of an virtual acoustics application can escalate the number of necessary spatialization resources. The time intervals between which resource assignments take place vary also widely between applications. Musical applications like the Helical Keyboard (or simulation of a clarinet using elementary sound sources) switch resource allocation with each note played. Teleconferencing applications would switch when new participants enter a shared space or the talker changes.

The spatialization might be done in hardware or on a separate system. Then the application requested CPU resources are not effected. Available systems (e.g., Intel RSX [Intel, Inc., 1997]) allow spatialization running on the same system as the application, enabled by simplification of the spatialization algorithms and general hardware improvements. In such a case an application might demand more CPU resources and the spatialization manager must reduce the number of spatialization channels to free CPU cycles. This process has to be done gracefully, minimizing acoustic artifacts that will irritate the user. Similar research is conducted in the field of multimedia (e.g., MPEG [Moser, 1996]).

The spatialization module might be in a server environment, providing service to more than one application. Also, with the introduction of multiple sinks the available spatialization channels vary. The spatialization resource manager must respond dynamically to such resource constraints.

If the computational costs for spatialization management are high, the win in better resource use might be lost or even worse.

The requirements may be summarized as:

- application adaptable;
- spatialization module/device flexible/independent;
- dynamic resource allocation/control;
- low computation costs compared to spatialization and application.

1.4 Outline

This thesis has the following structure. After the introduction of sound spatialization resource management in this chapter, sound spatialization management based on simple geometry is introduced in Chapter 2. The fundamental background is introduced in Chapter 3, starting with sound perception as a criteria for resource allocation, and also describing room simulation and reverberation, including early reflections, and the simulation and management of occlusion. An additional approach to reducing the necessary spatialization channels is optimizing through clustering, presented in Chapter 4. How applications can access the framework through the sound spatialization application programmer interface is presented in Chapter 5. The framework becomes portable by using an abstract spatialization backend interface, described in Chapter 6. The testing and calibration of spatialization backends is discussed in Chapter 7. The framework contains tools for resource allocation monitoring and visualization of the management process, presented in Chapter 8. How to use the framework for creating multimedia content is presented in Chapter 9. Finally, Chapter 10 concludes and extrapolates future trends.

Bibliography

[Aoki *et al.*, 1994] Shigeaki Aoki, Michael Cohen, and Nobuo Koizumi. Design and control of shared conferencing environments for audio telecommunication. *Presence: Teleoperators and Virtual Environments*, 3(1):60–72, 1994.

[Aritsuka and Hataoka, 1997] Toshijuki Aritsuka and Nobuo Hataoka. Gui representation system using spatial sound for visually disabled. In *ASVA'97 — Int. Symposium on Simulation, Visualization and Auralization for Acoustic Research and Education*, pages 415–420, Tokyo, Japan, April 1997. The Acoustical Society of Japan (ASJ).

[Arons, 1992] Barry Arons. A review of the cocktail party effect. *Journal of the American Voice I/O Society*, 12:35–50, July 1992.

[Astheimer, 1995] Peter Astheimer. *Sonifikation numerischer Daten für Visualisierung und Virtuelle Realität.* PhD thesis, Technische Hochschule Darmstadt, April 1995. (In German), ISBN 3-8265-1021-6.

[Barfield *et al.*, 1997] Woodrow Barfield, Michael Cohen, and Craig Rosenberg. Visual, Auditory, and Combined Visual-Auditory Displays for Enhanced Situational Awareness. *The International Journal of Aviation Psychology*, 7(2):123–138, 1997.

[Blattner *et al.*, 1992] Meera M. Blattner, Robert M. Greenberg, and Minao Kamegai. Listening to turbulence: An example of scientific audiolization. In *Multimedia Interface Design*, chapter 6, pages 87–102. ACM Press, 1992.

[Caussé *et al.*, 1992] René Caussé, J. F. Bresciani, and O. Warusfel. Radiation of musical instruments and control of reproduction with loudspeakers. In *International Symposium on Musical Acoustics*, Tokyo, 1992.

[Cohen and Herder, 1998] Michael Cohen and Jens Herder. Symbolic representations of exclude and include for audio sources and sinks: Figurative suggestions of mute/solo & cue and deafen/confide & harken. In *Virtual Environments 98*, pages 95/1–4, Stuttgart, June 1998.

[Cohen and Koizumi, 1995] Michael Cohen and Nobuo Koizumi. Audio Windows for Virtual Concerts II: Sonic Cubism. In Susumu Tachi, editor, *Video Proc. ICAT/VRST:* **I***nt.* **C***onf.* **A***rtificial Reality and* **T***ele-Existence/Conf. on* **V***irtual* **R***eality* **S***oftware and* **T***echnology*, page 254, Makuhari, Chiba; Japan, November 1995. ACM-SIGCHI (TBD), SICE (**S**ociety of **I**nstrument and **C**ontrol **E**ngineers), JTTAS (**J**apan **T**echnology **T**ransfer **A**ssociation), and NIKKEI (**Ni**hon **Kei**zai Shimbun, Inc.).

[Cohen and Koizumi, 1998] Michael Cohen and Nobuo Koizumi. Virtual gain for audio windows. *Presence: Teleoperators and Virtual Environments*, 7(1):53–66, February 1998. ISSN 1054-7460.

[Cohen, 1995] Michael Cohen. Besides immersion: Overlaid points of view and frames of reference; using audio windows to analyze audio scenes. In Susumu Tachi, editor, *Proc. ICAT/VRST:* **I***nt.* **C***onf.* **A***rtificial Reality and* **T***ele-Existence/Conf. on* **V***irtual* **R***eality* **S***oftware and* **T***echnology*, pages 29–38, Makuhari, Chiba; Japan, November 1995. ACM-SIGCHI (TBD), SICE (**S**ociety of **I**nstrument and **C**ontrol **E**ngineers), JTTAS (**J**apan **T**echnology **T**ransfer **A**ssociation), and NIKKEI (**Ni**hon **Kei**zai Shimbun, Inc.).

[Cohen, 1997] Michael Cohen. Exclude and include for audio sources and sinks: Analogs of mute/solo & cue are deafen/confide & harken. In *ICAD: Proc.* **I***nt.* **C***onf.* **A***uditory* **D***isplay*, pages 19–28, Palo Alto, CA, November 1997.

[Cohen, 1998] Michael Cohen. Quantity of presence: Beyond person, number, and pronouns. In Tosiyasu L. Kunii and A. Luciani, editors, *Cyberworlds*, chapter 19, pages 267–286. Springer-Verlag, 1998. ISBN 4-431-70207-5.

[Cook and Trueman, 1998] Perry R. Cook and Dan Trueman. A database of measured musical instrument body radiation impulse responses, and computer applications for exploring and utilizing the measured filter functions. In *International Symposium on Musical Acoustics*, Leavenworth, Washington State, USA, June 1998. Catgut Acoustical Society and Acoustical Society of America.

[Creative Technology Ltd., 1998] Creative Technology Ltd. Why did creative opt to support only 8 accelerated 3D streams? White paper, 1998. `http://www.sblive.com/livenews/3dstreams.html`.

[Dalenbäck *et al.*, 1996] Bengt-Inge Dalenbäck, Mendel Kleiner, and Peter Svensson. Auralization, virtually everywhere. In *the 100th Convention of the AES*, Copenhagen, May 1996. Preprint 4228 (M-3).

[Fouad *et al.*, 1997] Hesham Fouad, James K. Hahn, and James A. Ballas. Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. In Steven P. Frysinger and Gregory Kramer, editors, ICAD *'97 — Int. Conf. on Auditory Display*, pages 77–81, Palo Alto, CA; USA, November 1997.

[Goudeseune and Hamman, 1998] Camille Goudeseune and Michael Hamman. A real-time audio scheduler for pentium pcs. In *Proceedings of the 1998 International Computer Music Conference (ICMC 98)*, pages 158–162, Ann Arbor, October 1998.

[Hawley *et al.*, 1996] Monica L. Hawley, Ruth Y. Litovsky, Leah B. Dunton, Jennifer K. Jones, and H. Steven Colburn. Benefit of binaural hearing in a multi-source environment: Speech intelligibility. **J**. **A***cous.* **S***oc.* **A***mer.*, 99(4):2596, April 1996. 131st Meeting: Acoustical Society of America.

[Herder and Cohen, 1996a] Jens Herder and Michael Cohen. Design of a Helical Keyboard. In Steven P. Frysinger and Gregory Kramer, editors, ICAD *'96 — Int. Conf. on Auditory Display*, Palo Alto, CA; USA, November 1996.

[Herder and Cohen, 1996b] Jens Herder and Michael Cohen. Project report: Design of a helical keyboard. In *ICAD: Proc.* **I***nt.* **C***onf.* **A***uditory* **D***isplay*, Palo Alto, CA, November 1996.

[Herder and Cohen, 1997] Jens Herder and Michael Cohen. Sound Spatialization Resource Management in Virtual Reality Environments. In *ASVA'97 — Int. Symp. on Simulation, Visualization and Auralization for*

Acoustic Research and Education, pages 407–414, Tokyo, Japan, April 1997. The Acoustical Society of Japan (ASJ).

[Herder and Cohen, 1999] Jens Herder and Michael Cohen. The Helical Keyboard — Another Perspective for Virtual Reality and Music. *IJVR - International Journal for Virtual Reality*, 1999. In press.

[Herder, 1998] Jens Herder. Sound spatialization framework: An audio toolkit for virtual environments. *Journal of the 3D-Forum Society, Japan*, 12(9):17–22, September 1998.

[Huopaniemi *et al.*, 1996] Jyri Huopaniemi, Lauri Savioja, and Tapio Takala. Diva virtual audio relative system. In Steven P. Frysinger and Gregory Kramer, editors, ICAD*'96 — Int. Conf. on Auditory Display*, pages 111–116, Palo Alto, CA; USA, November 1996.

[Intel, Inc., 1997] Intel, Inc. Intel Realistic Sound Experience (3D RSX). White paper, 1997. `http://developer.intel.com/ial/rsx/WPAPER.HTM`.

[Karjalainen *et al.*, 1995] Matti Karjalainen, Jyri Huopaniemi, and Vesa Välimaäki. Direction-dependent physical modeling of musical instruments. In *15th International Congress on Acoustics (ICA'95)*, Trondheim, Norway, June 1995.

[Kashino and Hirahara, 1996] Makio Kashino and Tatsuya Hirahara. One, two, many — judging the number of concurrent talkers. **J**. **A***cous.* **S***oc.* **A***mer.*, 99(4):2596, April 1996. 131st Meeting: Acoustical Society of America.

[Kramer, 1994] Gregory Kramer. Some organizing principles for representing data with sound. In Gregory Kramer, editor, *Auditory Display*, volume XVIII of *SFI Studies in the Sciences of Complexity*, pages 185–221. Addison-Wesley, 1994.

[Martens and Myszkowski, 1998] William L. Martens and Karol Myszkowski. Psychophisical validation of the visible differences predictor for global illumination applications. In C. M. Wittenbrink & A. Varshney, editor, *Late Breaking Hot Topics Proceedings (IEEE Visualization '98)*, pages 49–52. IEEE Computer Society, October 1998.

[Martin, 1995] Keith Dana Martin. A computational model of spatial hearing. Master's thesis, Massachusetts Institute of Technology, June 1995.

[Meinard, 1997] Elizabeth D. Meinard. Creating auditory interfaces from graphical interfaces. In *ASVA'97 — Int. Symposium on Simulation, Visualization and Auralization for Acoustic Research and Education*, pages 123–126, Tokyo, April 1997. The Acoustical Society of Japan (ASJ).

[Moser, 1996] Martin Moser. Declarative Scheduling for Optimally Graceful QoS Degradation. In IEEE *Multimedia Systems '96*, Hiroshima; Japan, June 1996. IEEE.

[Ruspini and Khatib, 1998] Diego Ruspini and Oussama Khatib. Acoustic cues for haptic rendering systems. In *Proceedings of the third PHANToM User Group Workshop*, Dedham, MA, USA, October 1998.

[Savioja *et al.*, 1997] Lauri Savioja, Jyri Huopaniemi, Tapio Lokki, and Riitta Väänänen. Virtual Environment Simulation — Advances in the Diva Project. In ICAD*'97 — Int. Conf. on Auditory Display*, pages 43–46, Palo Alto, CA; USA, 1997.

[Todd *et al.*, 1994] Craig C. Todd, Grant A. Davidson, Mark F. Davis, Louis D. Fiedler, Brian D. Link, and Steve Vernon. AC-3: Flexible perceptual coding for audio transmission and storage. In *Proc. 96th Convention of the AES*, February 1994. Preprint 3796.

Chapter 2

Sound Spatialization Management

In a virtual reality environment, users are immersed in a scene with objects which might produce sound. The responsibility of a VR environment is to present these objects, but a practical system has only limited resources, including spatialization channels (mixels), MIDI/audio channels, and processing power. A sound spatialization resource manager controls sound resources and optimizes fidelity (presence) under given conditions. For that, a priority scheme based on psychoacoustics is needed. Parameters for spatialization priorities include intensity calculated from volume and distance, orientation in the case of non-uniform radiation patterns, occluding objects, frequency spectra (low frequencies are harder to localize), expected activity, and others. Objects which are spatially close together (depending on distance and direction) can be mixed. Sources that can not be spatialized separately can be mixed as ambient sources. Important for resource management is the resource assignment, i.e., minimizing swap operations, which makes it desirable to look-ahead and predict upcoming events in a scene. Prediction is achieved by monitoring objects' position, speed, and past evaluation values (i.e., priorities, probabilities, . . .). Fidelity is contrasted for different kind of resource restrictions and optimal resource assignment.

To give standard and comparable results, the VRML97 specification [Bell *et al.*, 1997] is used as an application programmer interface. Applicability is demonstrated with a helical keyboard [Herder and Cohen, 1996], a polyphonic MIDI stream driven animation including user interaction (a user may move around, playing together with programmed notes). The developed sound spatialization resource manager gives improved spatialization fidelity under runtime constraints. Application programmers and virtual reality scene designers are freed from the burden of assigning mixels and predicting the sound sources locations.

Spatial sound used in virtual reality environments fulfills different purposes. As passive feedback [Burdea and Coiffet, 1994, pp. 234-236] it enhance the realism of the display and informs the user about scene changes. In case of active feedback, the sound is directly coupled to user interaction (e.g., use of a manipulator, movement). Information presented include the position, orientation, movement, and speed of objects and the user in the scene. Besides that, room acoustic helps to understand the scene and get a feeling for space, walls, and materials. In this chapter, we mostly neglect room acoustics and focus on direct sound, which is important for localization of objects in space.

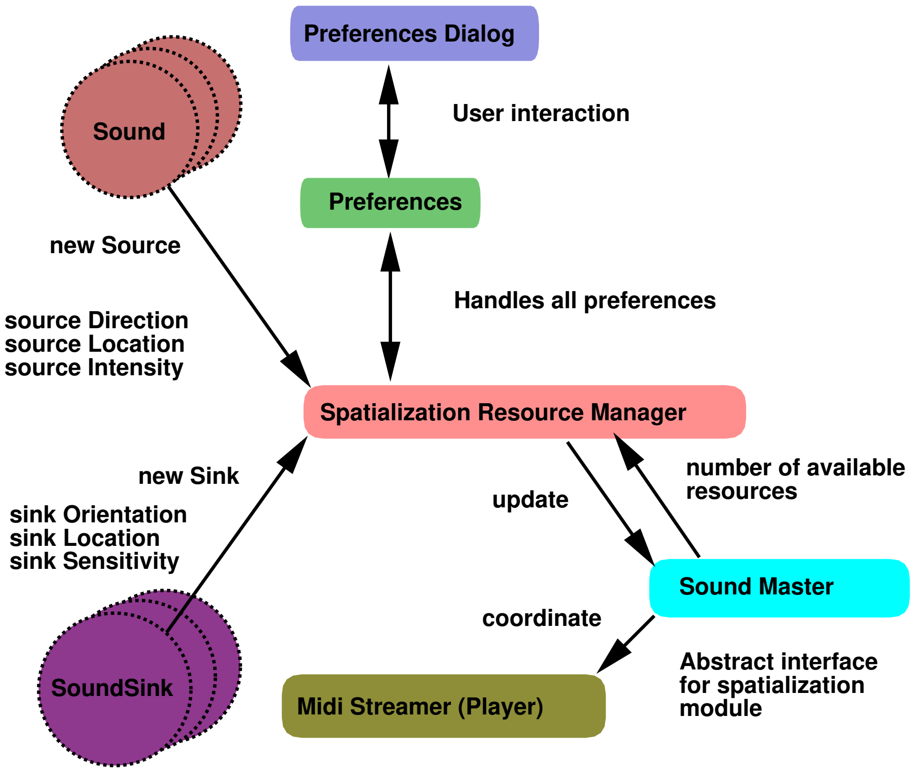


Figure 2.1: System schematic

Figure 2.1 shows a system schematic. We applied and tested the spatialization resource manager with two applications. In the Helical Keyboard project, shown in Figure 1.4 [Herder and Cohen, 1996], keys are objects in space which are activated by a MIDI stream. The number of requested simultaneous spatialized sources is defined by the polyphony of the song.

In the case of multiple sinks (see Section 2.2.3), the quantity of required resources (i.e., mixels) increases with the number of sinks. Each sink defines its own space, which is then is folded together with its siblings'. In a second application developed to show the capabilities of the resource management, objects follow a motion test pattern of distribution functions.

2.1 Requirements

Besides to the requirements given in the introduction (see page 13), the requirements for the algorithm introduced in this chapter are extended to:

- support for multiple sinks,
- using the API defined in Chapter 5, giving the application programmer easy control over resource assignment process,
- dynamic resource allocation/control, and
- low computation costs.

2.2 Resource management

Resource management can be static, in which the resource assignment is predefined by the VR scene designer, or dynamic, which means the mapping from source to channel is established at runtime. In systems in which the number of users and their sound spatialization requests are not predefined and the number of resources are limited, dynamic assignment of the resources will allow maximum system use.

What are sound spatialization resources? "Mixels,"— acronymic for '[sound] **mix**ing **el**ements,' in analogy to pixels, taxels (tactile elements), texels (texture elements), or voxels (a.k.a. boxels)— since they form the raster across which a soundscape is projected, define the granularity of control and degree of spatial polyphony.

Input (monaural) audio channels are associated with sources in the virtual space. Dynamic resource allocation assigns the source↔sink mappings to mixels, whose number is determined by the breadth of the directionalizing backend.

2.2.1 Strategies

A spatialization resource manager must decide how input channels are mapped to the resources. Figure 2.2 shows the principal task. Out of all relevant sources, sources for spatialization are selected. Our implementation is based mainly on the application programmer interface given by VRML 2.0 (see also Section 5).

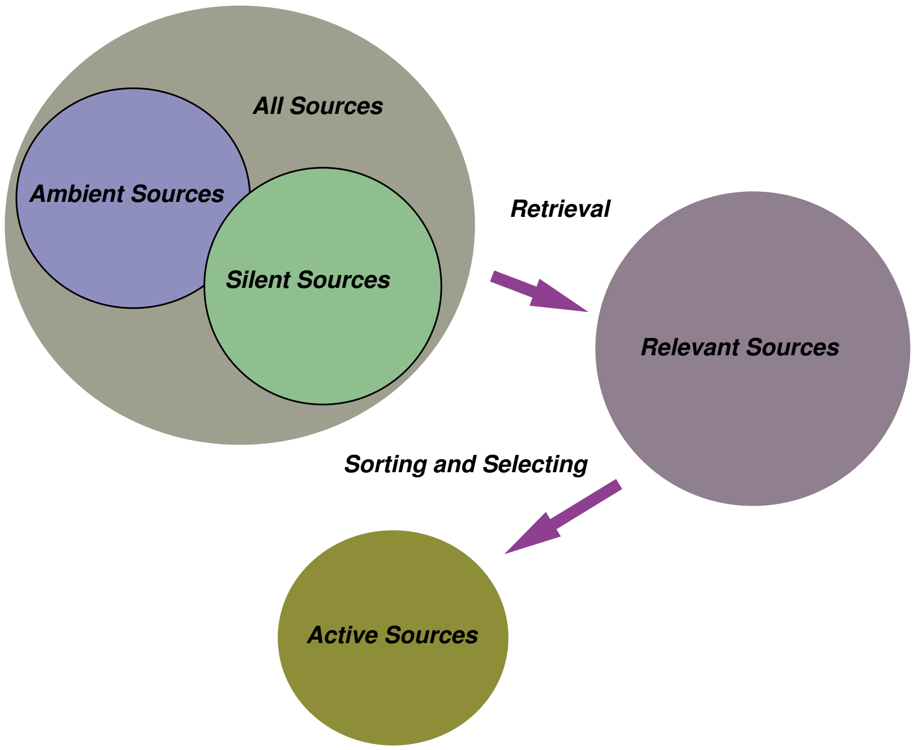


Figure 2.2: Source sets for spatialization

A simple algorithm would prioritize the relevant sources, assigning them to resources starting with the highest priority until no more resources were available. This operation might involve preemption of a source, which would be swapped out of the set of active sources. The following sections develop a more detailed and sophisticated approach.

Filtering relevant resources

Filtering Algorithm 1 is given in pseudo-code.

For simplicity the algorithm assumes an ellipsoidal radiation pattern of the sound sources and a spherical sensitivity pattern for sound sinks. The

Algorithm 1 Simple filtering algorithm

```
for each sink in sinks do
  if sink is enabled then
    for each source in sources do
      if source is ambient then
        if source is background then
          add source to background set of sink
        else
          add source to ambient set of sink
        end if
      else
        if source is low frequency then
          add source to ambient set of sink
        else
          distance ⇐ distance(source,sink)
          if ( not(sourceInSinkFarRange(source,sink) and
                 sinkInSourceAudibleRange(source,sink)) ) then
            add source to inactive set of sink
          else
            volume ⇐ f(intensity, sensitivity, distance)
            if ( sourceInSinkNearRange(source,sink) and
                 sinkInSourceCoreRange(source,sink) ) then
              volume ⇐ 1
            end if
            if volume < minVolume then
              add source to inactive set of sink
            else
              add source to active set of sink
            end if
          end if
        end if
      end if
    end for
  end if
end for
```

domain	trigger	operation	cost
multiprogramming/multitasking CPU	preemption, interrupt (OS timeslice)	swap out jobs (linked list of processes)	thrashing
virtual memory, caching RAM	page fault	swap out pages to disk	time
audio resource management	"interrupt" or reprioritization	swap out mixel to ambient space & spatialize new mixel	"fidelity," soundscape stability

Table 2.1: Resource management

function `f` defines the attenuation of sound in the medium and is used in Algoritm 1 and 2 for setting the volume value:

$$\mathtt{f}(intensity, sensitivity, distance) = \frac{intensity \times sensitivity}{distance^2} \tag{2.1}$$

Intensity and sensitivity are linearized and normalized gain values of source and sink, respectively.[1]

The boolean function `sourceInSinkFarRange` returns True if the source is within the far range sphere of the sink. Sources which are outside of this sphere are not audible to the specified sink. The field `farDistance` is field of sink controlled by an application.

$$\begin{aligned}\mathtt{sourceInSinkFarRange}(source, sink) = \\ distance(source, sink) < \mathtt{farDistance}\end{aligned} \tag{2.2}$$

The boolean function `sourceInSinkNearRange` returns True if the source is within the near range sphere of the sink. Sources which are outside of this sphere are attenuated or not audible to the specified sink. The field `nearDistance` is field of sink controlled by an application.

$$\begin{aligned}\mathtt{sourceInSinkNearRange}(source, sink) = \\ distance(source, sink) < \mathtt{nearDistance}\end{aligned} \tag{2.3}$$

Equation 2.4 defines when a sink is in the audible range of a source. The fields `maxBack`, `maxFront`, `location`, and `direction` specify an audible

[1] In this notation, "intensity" is not the physically defined term of acoustical power per unit area.

ellipsoid. If the sink is not inside of the ellipsoid, then the source is not audible by the sink. An ellipsoid is given by two focal points $f1$ and $f2$. Assume $f1$ is the location of the source node, then $f2$ can be calculated by $f1 + \texttt{direction}/|\texttt{direction}| * (\texttt{maxFront} - \texttt{maxBack})$. Let ls be the sink location, then the source is audible if $|ls - f1| + |ls - f2| \leq \texttt{maxBack} + \texttt{maxFront}$.

$$
\begin{aligned}
&\texttt{sinkInSourceAudibleRange}(source, sink) = \\
&\quad distance(sink{,}source) + \\
&\quad distance(sink{,}location(source) + \\
&\qquad\qquad\qquad \texttt{direction}/length(\texttt{direction}) * \\
&\qquad\qquad\qquad (\texttt{maxFront} - \texttt{maxBack})) \\
&\quad < \texttt{maxBack} + \texttt{maxFront}
\end{aligned}
\tag{2.4}
$$

A sink is in the core range of a source is defined in Equation 2.5 in the same way.

$$
\begin{aligned}
&\texttt{sinkInSourceCoreRange}(source, sink) = \\
&\quad distance(sink{,}source) + \\
&\quad distance(sink{,}location(source) + \\
&\qquad\qquad\qquad \texttt{direction}/length(\texttt{direction}) * \\
&\qquad\qquad\qquad (\texttt{minFront} - \texttt{minBack})) \\
&\quad < \texttt{minBack} + \texttt{minFront}
\end{aligned}
\tag{2.5}
$$

The values `minFront`, `maxFront`, `minBack`, and `maxBack` are attributes of the source, as defined by the API in Chapter 5, conforming to the Sound node definition [Carey and Bell, 1997, p. 277–284]; values `farDistance` and `nearDistance` are associated with the sink. Figure 2.3 shows the different ranges for sound source and sink as defined in Section 5.2 and Section 5.3. Source frequency is calculated by source channel, which in the case of MIDI is the normalized note number.

Sorting

For all active sources, as determined by techniques in this section, priority is calculated with Algorithm 2 based on volume, source and sink priority.

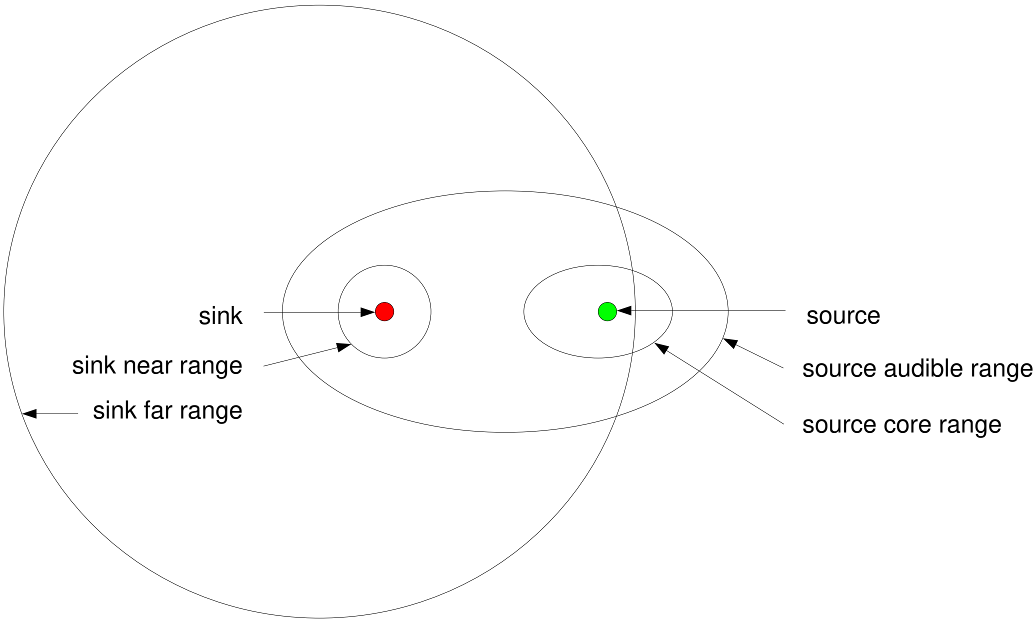


Figure 2.3: Audible and intensity ranges for sound source and sound sink

---

Algorithm 2 Simple algorithm to calculate source processing priority

---

for each sink in sinks **do**
  if sink is enabled **then**
    for each source in active sources attended by sink **do**
      distance ⇐ distance(source,sink)
      if $\left(\begin{array}{l}\text{sourceInNearRange(source,sink) and}\\ \text{sinkInSourceCoreRange(source,sink)}\end{array}\right)$ **then**
        volume ⇐ 1
      else
        volume ⇐ f(intensity, sensitivity, distance)
      end if
      source processing priority ⇐ volume * (sink priority + 1) * (source priority + 1)
    end for
  end if
end for

For efficiency, this calculation can be done together with filtering of relevant source. Sorting the set for priority and using the best for spatialization would already give a good strategy for resource allocation, but in the next section we optimize it further.

2.2.2 Reservation scheme

Resource reservation is available on three different levels. An application might reserve resources in advance via the API. In a spatialization server environment, the server might reserve resources. The spatialization resource manager reserves resources for sources which are currently not active but have shown in the past large values in probability in moving and high priorities. The past priorities (maximum and average) are set in contrast to the current active sources.

2.2.3 Multiple sinks

The spatialization resource manager must also consider multiple sinks [Cohen, 1995] [Herder and Cohen, 1996] which idiom was developed partly to address the issue of soundscape control in an shared context, allowing users to redirectionalize multiple sources without moving them (which might disturb other users) by installing and adjusting colocated sinks. An arbitrary number of users might experience a relaxed common view of a conference room or concert hall, each designating (possibly shared) sinks and using a personal gain adjustment profile, individually tuned for hearing acuity and **c**ontrol/**d**isplay characteristics.

Roles of sources and sinks A classification of sound sources and sinks is given in Table 1.1. Multiple sinks allow forked presence in auditory space. This is being like being in more than one place at once, a concept familiar from teleconferencing and studio recording. Two methods have been suggested for disambiguating the paradoxes of multiple presence. One is to partition the sources across the sinks, in which case the required mixels number is the same as a virtual space with a single sink. A second method crosses all sources and sinks, in which case required mixels number is the product.

2.3 Optimal sound spatialization manager

A spatialization module has a limited number of channels. An optimal spatialization resource manager would assign the channels to minimize the difference between a configuration with limited number of resources and an ideal one with unlimited resources.

2.4 Implementation

Our prototype was developed on an SGI Indigo 2 Extreme, connected with an Acoustetron II from Aureal/Crystal River Engineering and Roland Sound Module. The Open Inventor graphics toolkit was expanded for classes (nodes) to support the spatial sound extensions, which were used for our virtual reality applications. Open Inventor is a superset of the VRML 1.0 standard [Bell *et al.*, 1995], which does not support sound or dynamic behavior of objects.

Bibliography

[Bell *et al.*, 1995] Gavin Bell, Anthony Parisi, and Mark Pesce. The Virtual Reality Modeling Language, Version 1.0 Specification, May 1995. `http://www.vrml.org/Specifications/VRML1.0/`.

[Bell *et al.*, 1997] Gavin Bell, Rikk Carey, and Chris Marrin. ISO/IEC 14772-1:1997: The Virtual Reality Modeling Language (VRML97), 1997. `http://www.vrml.org/Specifications/VRML97/`.

[Burdea and Coiffet, 1994] Grigore Burdea and Philippe Coiffet. *Virtual reality technology.* Hermes, 1994. ISBN 0-471-08632-0.

[Carey and Bell, 1997] Rick Carey and Gavin Bell. *The Annotated* VRML *2.0 Reference Manual.* Addison-Wesley Developers Press, 1997. ISBN 0-201-41974-2.

[Cohen, 1995] Michael Cohen. Besides immersion: Overlaid points of view and frames of reference; using audio windows to analyze audio scenes. In ICAT/VRST*: Int. Conf. Artificial Reality and Tele-Existence/Conf. on Virtual Reality Software and Technology*, pages 29–38, Makuhari, Chiba; Japan, November 1995.

[Herder and Cohen, 1996] Jens Herder and Michael Cohen. Design of a Helical Keyboard. In Steven P. Frysinger and Gregory Kramer, editors, ICAD *'96 — Int. Conf. on Auditory Display*, Palo Alto, CA; USA, November 1996.

Chapter 3

Sound Perception and Room Effects

In an information-rich virtual reality environment, the user is immersed in a world containing many objects providing that information. Given the finite computational resources of any computer system, optimization is required to ensure that the most important information is presented to the user as clearly as possible and in a timely fashion [Herder and Cohen, 1997a].

3.1 Conceptual model

A conceptual model, diagrammed in Figure 3.1, has three layers. The top layer spans the known human-machine interface. A middle layer distinguishes between perceivable space/sensor system and motor system/feedback [Sanders and McCormick, 1987, p. 46–47] [MacKenzie, 1995, p. 438]. The models layer completes the classification. Typical user interfaces do not consider in a dynamic way the perceivable space of the user; only at the design stage is this estimated, and it is statically frozen by programming. More advanced user interfaces measure user performance, or can switch between different modes to support the users. Examples of such adaptation include classification of user expertise and then offering more functions in menus and less or different help texts [Sukaviriya and Foley, 1993]. Research in adaptive user interfaces has been done with focus on various aspects including adaptive automatic display layout [Stille *et al.*, 1996] and information filtering for pilots based on workload [Mulgund and Zacharias, 1996].

A virtual reality user interface can provide more information to a user than WIMP-based interfaces by using broader sensory channels. Also, more information about an immersed user can be obtained by the system and used

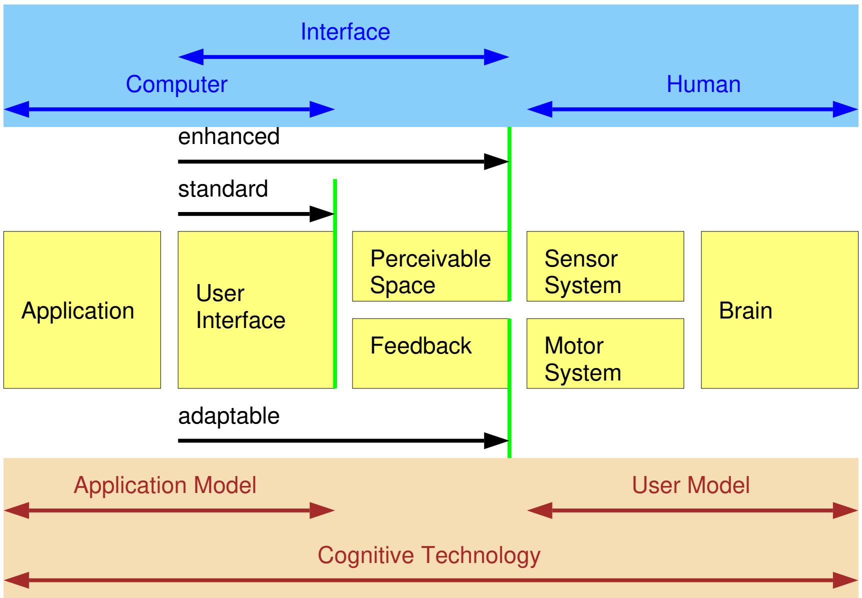


Figure 3.1: Concept — Interface — Models

to enhance the user interface in a dynamic fashion. Examples are head- or eye-tracking for finding out users' focus area.

3.2 Resource management

Human interface resources can be classified using different taxonomies. One such organization classifies according to resources provided by the computer. Another is to classify by the ability of the user to perceive information [Ovan and Havens, 1993].

It is not necessary to compute (i.e., use system resources) display data which the user cannot perceive because of occlusion, masking, or low level. Rendering devices have limited capabilities. A sound spatialization backend (e.g., Acoustetron) can render (i.e., put monaural sound sources into three-dimensional space) only a limited number of mixels (e.g., eight channels) simultaneously. The responsibility of the resource manager is to determine the set of computable renderable data $R_{\text{computation}}$, which is a subset of both $R_{\text{displayable}}$ and $R_{\text{perceivable}}$.

$$R_{\text{computation}} \subseteq R_{\text{displayable}} \cap R_{\text{perceivable}}$$

$R_{\text{displayable}}$ is the set of resources which are available for the display.

$R_{\text{perceivable}}$ is the set of resources which the user can perceive. At a given point in time, $R_{\text{computation}}$ is optimal if there is no larger set which fulfills the requirements. This is not necessarily the best solution over a long period in time because allocated resources cannot be freed immediately, so that in a dynamically changing scene, a non-optimal solution for a short time period could give, on average, a better impression.

3.3 Audio resources

The responsibility of the audio module in a VR environment is to present sound sources as well as possible, but any practical system has only limited resources, including spatialization channels (mixels), MIDI/audio channels, and processing power. A sound spatialization resource manager [Herder and Cohen, 1997b] controls sound resources and optimizes fidelity (presence) under given conditions. For that, a priority scheme based on psychoacoustics is needed. Parameters for spatialization priorities include:

- intensity, calculated from volume and distance,
- orientation, for non-uniform radiation patterns,
- occluding objects,
- frequency spectra (low frequencies are harder to localize), and
- expected activity.

Objects which are spatially and perceptually close together (depending on distance and direction) can be clustered. Sources that cannot be spatialized separately may be mixed as ambient sources.

3.4 Room simulation

Localization means not only directionalization, whence a source emits, but also range estimation and a "feeling" of the space. The auditory environment gives a context to the situation, the space, and helps to orient the user in it.[1]

Head-related transfer functions are usually measured only at a fixed distance. The convolution of a low reverberant signal with HRTFs gives good

[1] The reverberation missing in an anechoic chamber does not allow usual localization just by using the auditory senses. People in a dark anechoic room experience a kind of "lost in space" sensation which can even approach fear.

	spatialization	localization
media	hardware/software	auditory senses
function	directionalization (i.e., θ, ϕ) distance	orientation
space	room effects	presence

Table 3.1: Spatialization/localization taxonomy

directional cues but not distance cues. It is necessary to add room reverberation to the processed signal [Anderson and Casey, 1997]. Such a processing scheme is shown in Figure 3.2.

Role of reverberation in distance perception In visual perception, if one sees a larger object far and a smaller object near, their relation can be recognized from context [Ishikawa *et al.*, 1998]. Aural perception includes analogous effects. The magnitude of a sound is source intensity, having a physical, objective value. But perceived source level is loudness, a mental percept. If a source with small intensity is close to a listener, it is recognized as such through reverberation, which is analogous to the context (objects of known size) in visual perception. Since reverberation is the environmental context, it must be constant across distance, depending only on source intensity.

3.5 Audio rendering based on an image model

A common technique for including early reflections in audio rendering is to use an image model [Begault, 1994, p. 184] [Kendall and Martens, 1984]. An extension of this is the cellular approach for modeling room acoustics [Dhillon, 1994]. For the rendering two passes are necessary. In a first pass, rays are sent out from sound sources to the sound sinks. This process is analogous to ray-tracing in graphics. From those rays, representatives are chosen using a selection process. Along those significant rays, filter functions are calculated for radiation, reflection, and occlusion. In a second pass, the filter functions are applied to the audio signals. As optimization, the filter function can be compiled to one filter function. Figure 3.3 shows the image-based rendering process. The filter functions require local information. The radiation function is based on radiation direction and radiation pattern. A reflection filter function uses material data, surface normal and incoming direction. The occlusion function [Tsingos and Gascuel, 1997] uses the ratio

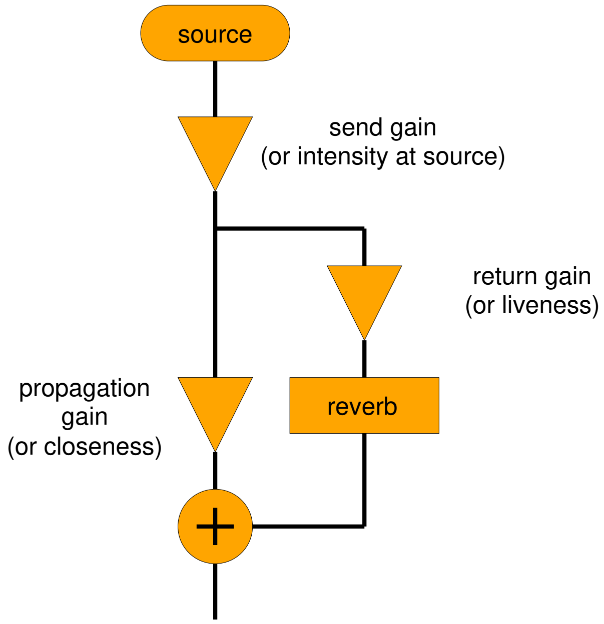


Figure 3.2: Source loudness amplification

between room volume and occluder volume. Part of the necessary data can be provided in a system with integrated graphics and audio rendering by the graphics process.

The complexity of sound rendering based on ray-tracing is illustrated in the following paragraph. Equation 3.1 shows the calculation of a signal along one ray s_{r_i} by applying the sink filter function L_k, reflection and occlusion filter functions F_{h_i}, and source radiation filter function R_i to a source signal s_i. The index h_i runs from 1_i to n_i. The cardinal n_i depends on the number of occlusions and reflections as well on the maximum numbers of filter functions, determined by system capabilities or user tuning, for one ray (usually limited). A signal of a source reaches the sink along different paths (rays). Therefore the signal contributed by a source s_{o_j} is given in Equation 3.2, which is the summation over all rays. The calculation of a signal for an output channel s_k is given in Equation 3.3 by adding the signal for room reverberation and contributions of all sound sources. For a headphone based system the range for k is two.

$$s_{r_i} = L_k * F_{1_i} * \ldots * F_{n_i} * R_i * s_i \quad (3.1)$$

$$s_{o_j} = \sum_{i \in rays} s_{r_i} \quad (3.2)$$

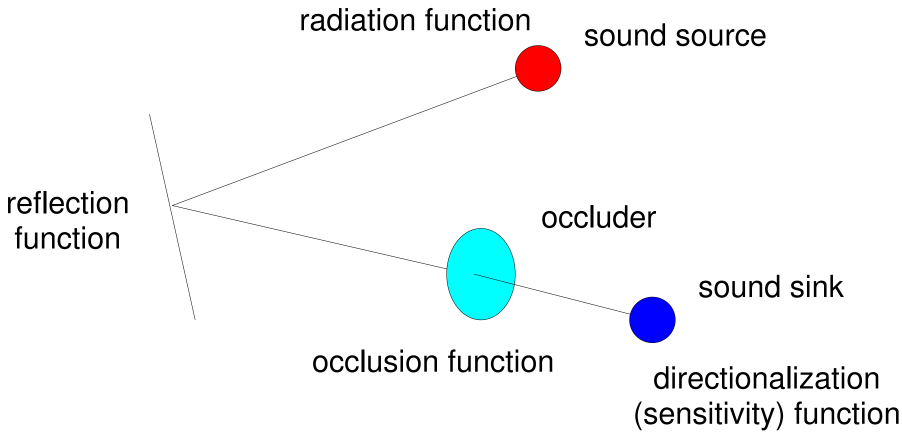


Figure 3.3: Image-based rendering

$$s_k = s_r + \sum_{j \in sources} s_{o_j} \tag{3.3}$$

The following table lists the used symbols for calculating the output signal:

R	source signal radiation function
F	reflection or occlusion filter function
L	sink filter function (for directionalization; in case of headphones: HRTF)
k	output channel
s_i	raw source input signal
s_{r_i}	signal along one ray
s_{o_j}	signal contribution for a source
s_k	signal for an output channel
s_r	signal for room reverberation

3.6 Early reflections

Early reflections, especially first-order reflections, contribute to localizability and space awareness. First- and higher-order reflections can be simulated using source an image model [Wenzel, 1994] based on mirror-like symmetries. First- and second-order reflections based on the source image model have been implemented in a spatial reverberator [Kendall and Martens, 1984, p. 118–120]. The image model assumes specular reflections and practical realtime systems limit the order to two or three [Begault, 1994, p. 184–186].

Implementing reflections using the image source model like that mentioned in the previous section does not scale well for complex environments.

A very promising approach [Funkhouser *et al.*, 1998] based on beam-tracing[2] depends mostly on the complexity of the local environment. Currently, this approach works only for fixed source locations, but extensions are expected.

3.7 Perceptual space

Section 3.1 introduced the notion of a sensory space of the user giving a global context, but did not explore it further. For spatialization resource management (Chapter 2), criteria are necessary to make the allocation and clustering (Chapter 4) decisions. In [Herder and Cohen, 1997b], application programmers' priority was used as criteria, coupled with a simple physical model based on distance and intensity. This does not adequately capture the capabilities of the human auditory system, which has different abilities to perceive sound depending on environment and direction. This richness is well-known, often measured using audible limens, also known as "just noticeable differences" (JND) [Williams, 1994, p. 100] [Blauert, 1996, p. 16]. A similar term is "minimum audible angle" (MAA) [Begault, 1994, p. 40]. This space is inhomogeneous. If distances based on Euclidean space are used as criteria for sound spatialization management, then errors are introduced as the perceptual capabilities of a human listener are not considered. A mapping from the Euclidean space of the virtual reality environment (object space) to perceptual space, which equalizes the audible limen, can solve this problem. In this space all decisions concerning spatialization resource management, like relevance calculation and clustering, are done. From this space we can get back into the object space by keeping references to the objects. This is more efficient and accurate than trying to define a function which would do the back-mapping. This perceptual space is context- and application-dependent. For example, head-tracking changes auditory resolution.

The perceptual space is used for answering following important questions of the resource management:

- Is a given audio event/signal audible[3]?
- Is a given audio event/signal masked by another signal?
- Is a given audio event/signal localizable separately from another audio event/signal?

[2] a beam covers a collection of rays

[3] "Audible" here means that the signal is louder then the noise/reverberation level of the system.

- Does a first- (or higher-) order reflection contribute to the perception of a scene?
- Does an occluder contribute to the perception of a scene?

3.7.1 Construction of the perceptual space

Head-related transfer functions (HRTFs) are measured and are in common use for sound spatialization. HRTFs can also be used to calculate the intensity at the ear for a given signal with a certain frequency range at a certain source location relative to the listener position.[4] Other studies suggest loudness and audibility functions dependent on frequency [Schiffman, 1995, p. 351–353] [Blauert, 1996, p. 120]. Thresholds for audibility are insufficient for determining actual audibility because the signal might have a level lower than the room noise.

Besides the question of audibility, an important factor is localization blur, which depends on frequency and orientation of the listener [Blauert, 1996, p. 40–50]. Multiplying the values of both functions gives loudness dependent of distance, orientation (i.e., location), and frequency. The resulting function can be used for guessing the audibility and maskability of audio events. The localizability has be modeled separately.

3.8 Role of the frequency band in directionalization

The frequency band of a sound source plays an important role in the ability of an user to perceive a sound source from a certain direction. A listener is able to differentiate sound sources in the horizontal plane using interaural time difference (ITD) and interaural level difference (ILD). The cone of confusion [Handel, 1989, pp. 110–111] [Blauert, 1996, p. 179], shown in Figure 3.4, describes the locus of points with the same ITD and ILD. Spectral differences allow listeners to discriminate sound sources on the cone of confusion. The spectral content of a sound is correlated how well a sound source can be spatialized. Even for narrow-band signals, predictions of ability to locate a sound source can be made [Blauert, 1968].

Table 3.2 shows qualitative relations between frequency band and ability to directionalize sound sources. Telephone equipment used for voice communication have cutoff frequency at around 5 kHz which reduces the required bandwidth.

[4] This is a convolution of the signal with the HRTF for the specified location.

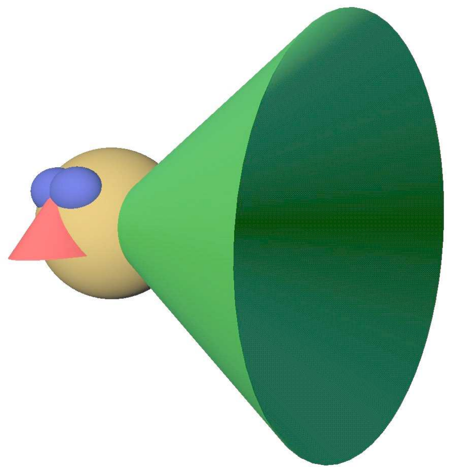


Figure 3.4: Cone of confusion; locus of points with the same ITD and ILD

frequency range	application	front ↔ back confusion	elevation discrimination
low	voice, conferencing	high	low
high	music, environmental sounds	low	high

Table 3.2: Frequency band determines limits of spatial resolution

Harmonic distortion and interaction with other objects can generate higher frequencies.

3.8.1 Localization error depending on target direction

Figure 3.5 shows horizontal and vertical unsigned localization errors for a broadband signal (measured values are taken from [Makous and Middlebrooks, 1990]). The ellipses axes of the cones denote error in azimuth and elevation. A generalizing of the data suggests that the error in azimuth to the front is small, growing larger to the sides, while error in elevation decreases at the sides. The back has much higher error than the front.

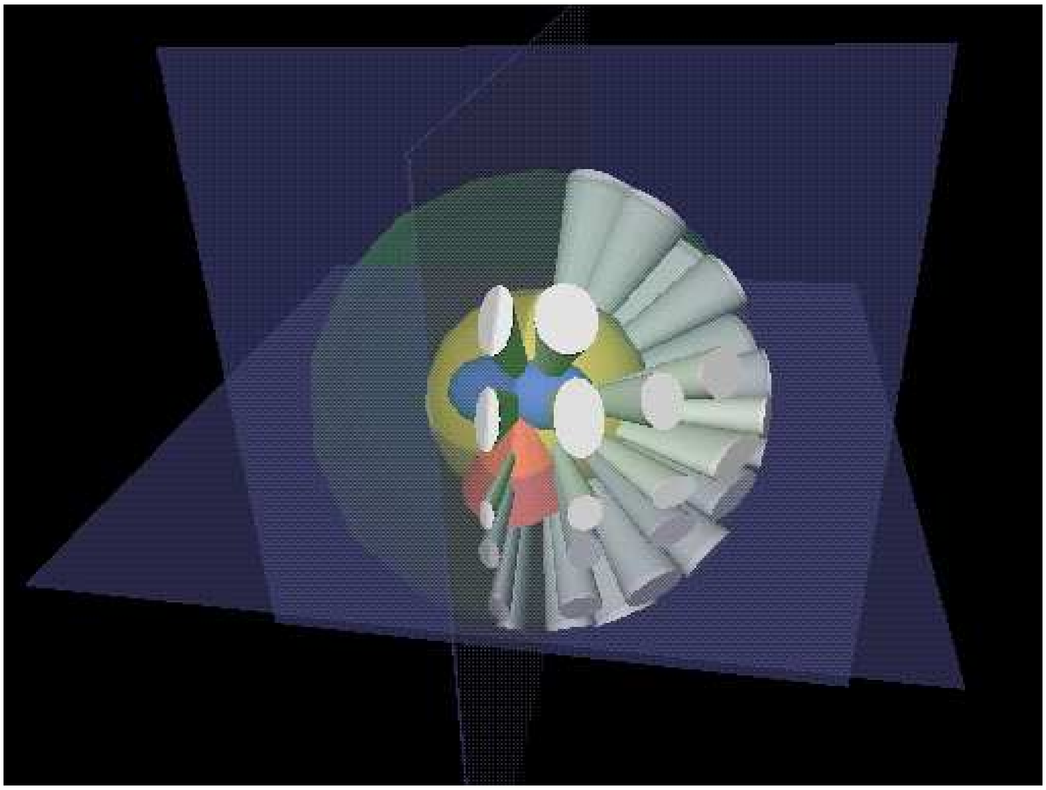


Figure 3.5: Horizontal and vertical unsigned localization errors for a broadband signal; ellipses axes denote error in azimuth and elevation

3.8.2 Elevation discrimination

In a listening experiment at the University of Aizu [Suzuki, 1999], elevation discrimination was investigated for two types of stimuli. Voice samples had a bandwidth of 5 kHz and an explosion sample had a bandwidth of 10 kHz. The experiment was done with 5 subjects in an anechoic chamber for five elevation angles and four major azimuth angles. The influence of reverberation on an identification task was studied by presenting reverberation through two separate loudspeakers.

The results are summarized in Figure 3.6. Sensitivity is shown using the d' metric [Green and Swets, 1966, p. 405], the difference of the Z distribution of the hit rate and false alarm rate, which excludes biases. Elevation angles were better estimated without reverberation. At lateral angles the identification performance was better than in the front or back. High frequency stimuli could be better identified.

3.9 Sound occluder

In previous research [Tsingos and Gascuel, 1997], Fresnel zones[5] had been used to calculate a filter function of a given occluder along a sound path. Visibility methods like those used in computer graphics allow efficient calculation [Tsingos and Gascuel, 1997]. In other work, an occlusion volume was suggested [Ellis, 1998] as an extension for the VRML97 specification [Bell *et al.*, 1997]. This newly proposed `SoundOcclusion` node represents the volume as a sphere that defines an infinite impulse response filter (IIR).

In the interest of improving audio rendering efficiency, a simplified filtering model was developed [Martens *et al.*, 1999]. Two perceptually salient components of occluder acoustics were identified that could be directly related to the geometry and orientation of a simple occluder. Actual occluder impulse responses measured in an anechoic chamber resembled the responses of a model incorporating only a variable delay line and a first-order (one-pole, one-zero) filter. The filtering model has the attractive feature that either the reflecting or the occluding effects of an obstruction can be simulated through changes in continuously variable parameters, requiring no change in the filter structure.

What does acoustical occlusion sound like? When an object is interposed between a source and a sink, and that object is not acoustically transparent, it typically reduces the sound pressure level (SPL) that can be measured at the sink position. It also changes the spectral energy distribution, and

[5]Fresnel zones are volumes enclosed between ellipsoids.

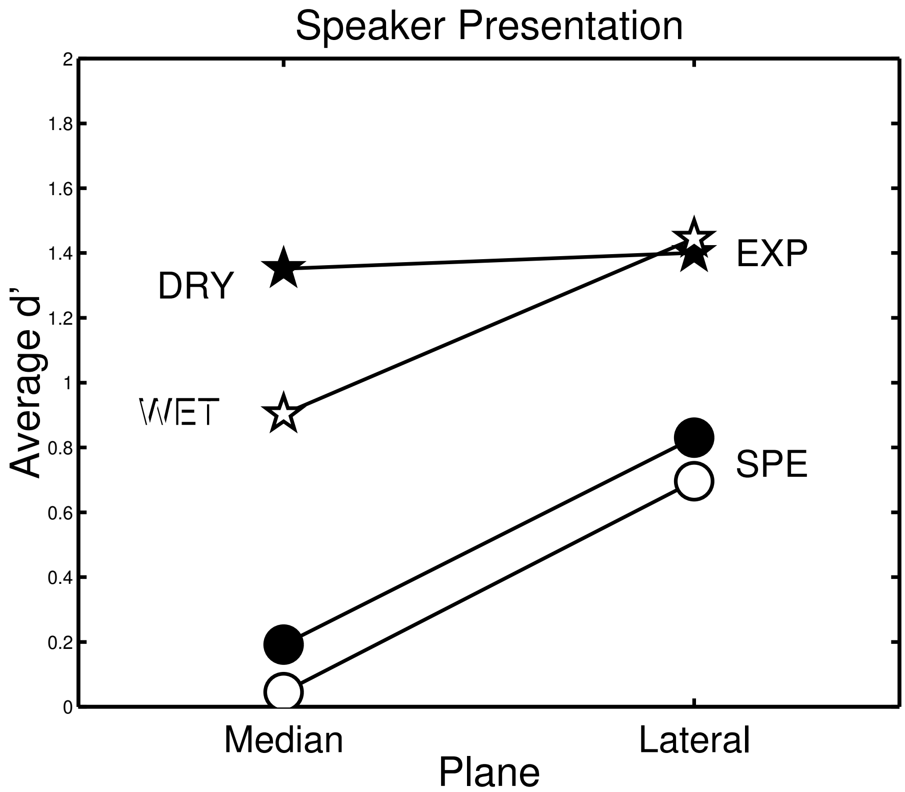


Figure 3.6: Sensitivity for elevation discrimination depends on content and direction: d' measures the sensitivity; stars (EXP) are the results for the explosion samples; circles (SPE) are the results for speech samples; full symbols (DRY) are results without reverberation

usually smears the energy over time as well. But when are the changes to a sound source audibly characteristic of occlusion? In Section 3.10 this general question was reduced to the specific experimental question, “When can a human listener confidently detect the presence of an occluder in a blind listening test?” Because the SPL at the source was varied from trial to trial, the SPL at the receiver position was an unreliable cue for the detection of occlusion in that test. Under these circumstances, occlusion effects were somewhat difficult to identify. One goal was to develop a filtering model that could effectively synthesize signals that would sound more clearly as if they resulted from occlusion. Another goal was to develop efficient means for rendering the spatial image of an occluded virtual sound source in order to minimize computational cost while maximizing effect audibility.

In contrast to the goal of reducing rendering cost by selectively eliminating reflectors and occluders from the rendering equation [Martens and Herder, 1999], the motivation was to simplify the rendering algorithm, and thereby allow more sonic effects to be rendered.

An alternative approach to rendering the complex spectro-temporal phenomena associated with occlusion is to reduce the effects to simple changes in sound source gain. This approach was taken by [Takala and Hahn, 1992], who set gain via a scaling factor proportional to the amount of occlusion. What is the rationale for such gross simplification? In many multimedia applications, as well as in conventional audio production for sound effects, the purpose of audio rendering is to illustrate an event rather than to attempt an acoustically accurate simulation [Takala and Hahn, 1992]. Given that occlusion effects can be difficult to identify in blind listening tests [Martens and Herder, 1999], there seems to be some justification for such extreme rendering simplifications. On the other hand, there are changes in tone color of the direct sound that are characteristic of some occlusion effects. In contrast to highly detailed physical solutions, however, such as those based upon Fresnel zones [Tsingos and Gascuel, 1998], a solution was desired that was driven rather by perceptually-defined specifications. As a more detailed alternative to the simplification that reduces occlusion effects to changes in gain, this section presents a filtering model of low computational cost that can produce distinctive auditory spatial images associated with identifiable occlusion effects.

First, many acoustical measurements of an actual occluder were made in an anechoic chamber in order to characterize such stimuli. Second, a digital filtering model was designed to capture the variation in occluder impulse responses that was observed as the occluder size, position, and orientation were varied.

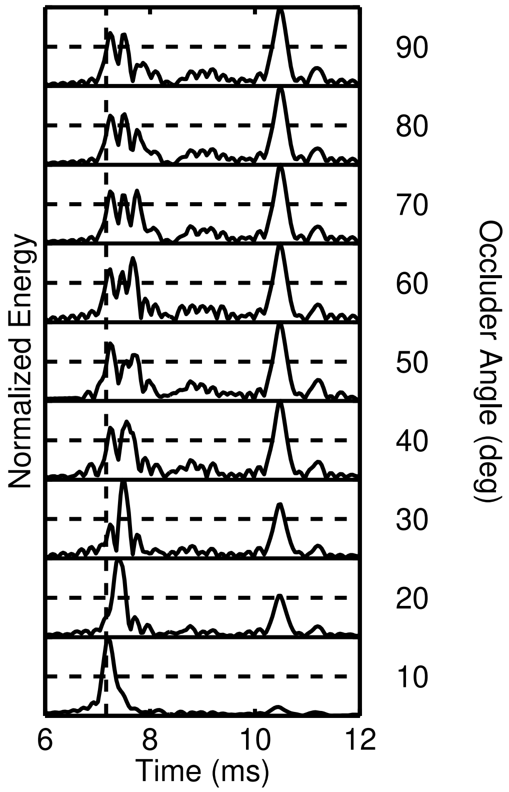


Figure 3.7: Time-domain responses at nine occluder angles

3.9.1 Acoustical measurement of occlusion

The acoustical responses of obstructions of various sizes were measured at various positions and orientations relative to the location of a loudspeaker and a microphone. Figure 3.7 shows the changes in the response of a small, rectangular board of 40 by 90 cm as its orientation varied in relation to a sound source and sink that were each located 100 cm from the center of the board. The curves plot normalized energy over time for the response at nine occluder angles. The 90° angle provided the maximum obstruction to sound in this set of measurements, while the 10° angle presented the smallest silhouette along the line of sight between source and sink. The vertical dashed line marks the time at which the normalized energy peaks for an unobstructed sound arriving at the microphone position via direct path from the loudspeaker. Note that the amplitude of the initial peak observed at around 7 ms decreases with occluder angle while that of the secondary peak observed at around 10.5 ms increases with occluder angle (with increasing obstruction). The absolute peak energy is also declining with elevation, though that detail is obscured in the figure by normalizing energy for each measurement according to the maximum of each curve.

The magnitude response over frequency for the nine occluder angles (not shown here) showed substantial attenuation in a broad band centered at 5 kHz, but showed virtually no attenuation around 8 kHz. At 10°, very little attenuation was observed at frequencies below 8 kHz. The small, rectangular occluder almost always shows a boost in gain at frequencies ranging from .1 to 1 kHz, even at a 20° angle that presents a very small silhouette along the line of sight between microphone and speaker.

3.9.2 Filter design for occlusion and reflection simulation

The filtering model was designed to capture the sonic effects of a modeled obstruction, whether the sound source is reflected or occluded by that obstruction. Because obstructions located near the direct-sound path are always switching between these two states, a comprehensive solution was required. The filter operates by tapping the audio input buffer containing the sound-source audio samples either earlier or later than the tap for the direct-sound signal. If the obstruction is oriented as a reflector, then the obstruction signal is delayed relative to the direct sound, and receives a high-pass emphasis. If, however, the obstruction is oriented as an occluder, then the obstruction signal will lead the direct sound, and is given a low-frequency emphasis.

Characteristic magnitude responses resulting when obstruction signals are

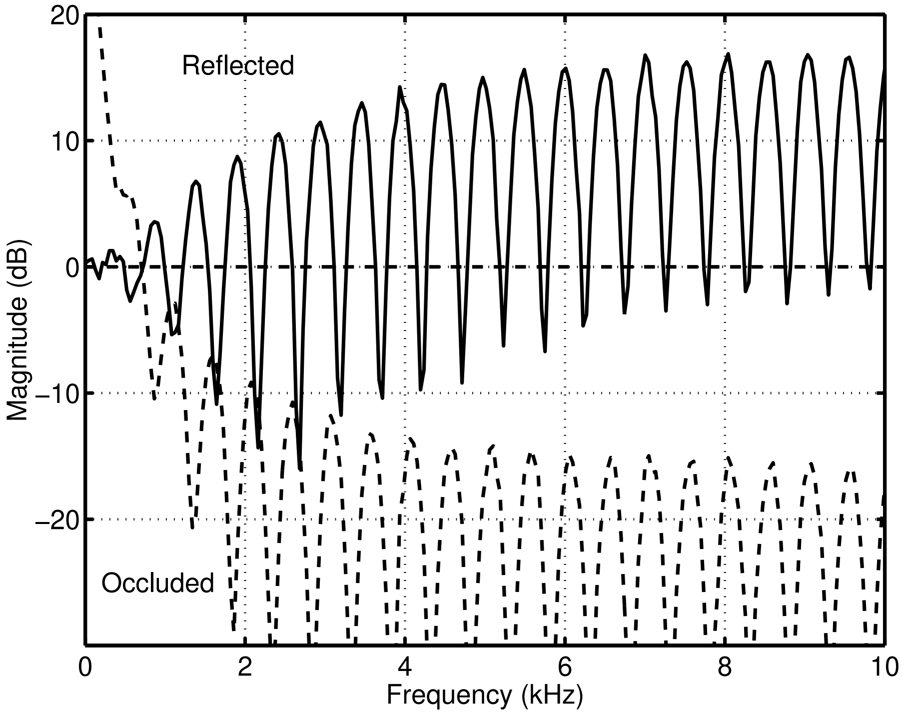


Figure 3.8: Filter model magnitude response for reflected and occluded sound

combined with direct-sound signals are shown in Figure 3.8. For the relatively small obstruction simulated here, the upper, solid curve shows that the high-frequency portion of the direct sound is reflected at higher gain. In contrast, the lower, dashed curve shows that the low-frequency portion of the occluded direct sound is amplified in the model, somewhat exaggerating the increase in gain observed in measured responses. In both cases, the depth of the comb pattern in the magnitude response increases with increasing frequency.

Attenuation of DC energy for early reflection simulation is common [Kendall *et al.*, 1986], but a single filter structure that can smoothly transform reflection effects into occlusion effects is novel and advantageous.

The filtering model described here allows for smooth transition from occluder to reflector state, easing processing control by handling the two states between which obstructions are always switching. Perceptual evaluation confirmed that the model enables the creation of a continuous range of obstruction effects that also produces satisfying auditory spatial imagery.

A criteria for removing an occluder from a rendering process is suggested in the next section.

3.10 Perceptual criteria for eliminating reflectors and occluders from the rendering of environmental sound

This section discusses determination of effective means for choosing which components of the rendering of a reflector or a occluder would provide the most audible differences for spatial sound imagery. Rather than begin with an analytic approach that attempts to predict audible differences on the basis of objective parameters, subjective tests of how audibly different the rendering result may be heard to be when that result includes two types of sound obstruction (reflectors and occluders) give an approximate answer. Single-channel recordings of 90 short speech sounds were made in an anechoic chamber in the presence and absence of these two types of obstructions, and as the angle of those obstructions varied over a 90° range (see Figure 3.9).

In two listening experiments, these recordings were reproduced over a single loudspeaker in the anechoic chamber, and listeners were asked to rate how confident they were that the recording of each of these 90 stimuli included an obstruction. In a parallel experiment, listeners recorded confidence ratings for the presence of an occluder when the stimulus presentation included simulated reverberation. Because the sound source varied from trial to trial in these experiments, confidence ratings were scattered, but clearly modulated

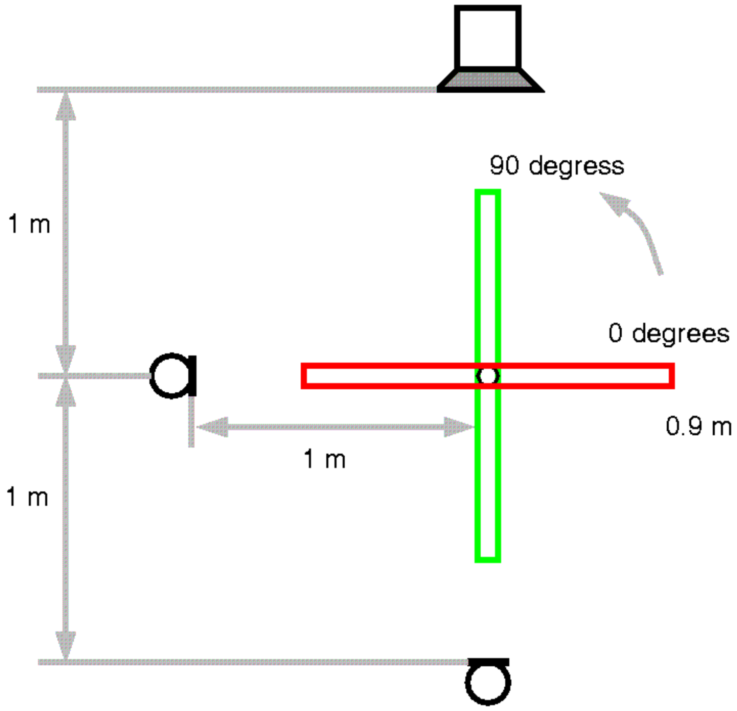


Figure 3.9: Reflector and occluder recording setup

by obstruction parameters. Although obstruction identification was already difficult under anechoic listening conditions, embedding the test signal in simulated reverberation did not reduce identification performance. Thus, at least for obstruction effects that operate within the first few milliseconds of direct sound arrival time, the obtained confidence ratings provide a basis for an evaluation function that predicts which reflectors and occluders might be most important for rendering.

Resource management in the rendering of spatial sound [Herder and Cohen, 1997b] is important for efficient synthesis of realistic sound imagery. Complex environments can contain many sonic obstructions, including objects that occlude direct sound (interposed between sound source and sink), and objects that reflect sound (producing an additional sound signal that delays, attenuates, and combines with the direct sound traveling from source to sink). Of course, some of these sonic effects are not significant, but predicting which will be perceptible is a difficult problem [Begault, 1996]. Besides the indirect sound associated with the walls, ceiling, and floor of a modeled space, spatial audio rendering for complex environments must include considerations of components for objects located within the enclosure, such as reflectors and occluders. In contrast to enclosure-related reflections that arrive at the listening position with longer delays, the contribution of such sonic obstructions to auditory spatial image formation is predominantly focussed on characteristics of the source, such as tone coloration [Barron, 1971].

Two questions were addressed in the experiments reported here. The first question concerns the audibility of obstruction effects when correlated parameters such as loudness are unreliable cues to the presence or absence of an obstruction: Are obstruction effects *per se* identifiable in blind listening tests? The second question concerns the potential masking effects that subsequently arriving indirect sound might have on the audibility of obstruction effects: Are obstruction effects less identifiable when the presentation includes (simulated) reverberation?

3.10.1 Method

For the first two experiments, a constructed set of 360 stimuli for four separate listening sessions was used for testing, in which the angle of an obstruction (i.e., reflector or occluder) was varied over a 90° range. The presence or absence of these obstructions was randomly varied from trial to trial, there being a 50% probability that a given trial would contain only the unobstructed direct sound. The stimuli were presented in an anechoic chamber over a small loudspeaker located directly ahead of the listener in a distance of 230 cm. The sound source used was also different for every trial, taken from a set of anechoic speech samples (Japanese numbers ranging from 1 to 90) that had been recorded by one of five human speakers. The listening sessions were conducted separately. In each listening session of 90 stimuli, the initial angle of the obstruction was zero degrees (which means perpendicular incidence of the direct sound on the obstruction surface). In order to minimize uncertainty, and thereby maximize the likelihood of correct identification of the obstruction, the obstruction angle was changed gradually from trial to trial. So in each successive trial, the obstruction was rotated by one degree relative to the previous trial's obstruction angle. Thus the 45th trial contained a reflector angle that most closely provided a specular reflection for the sound source (i.e., the angle formed by incident and reflected sound rays was split by the surface normal of the reflector). In the second and final listening session under each condition, the complementary set of trials was presented. For example, if in the first session a reflection was presented in the 10th trial at a reflection angle of 10° , then in the second session the dry source would be presented in the 10th trial. Listeners were asked to estimate the presence or absence of an obstruction using a scale from 1 to 5. If they were confident that the obstruction was present, then the response category reported was "5." If they were confident that the obstruction was absent, then the response category reported was "1."

For the third experiment, a unique sound source was again employed on each trial; however, only four obstruction configurations were included in the stimulus set. Here again, listeners rated their confidence that an occluder had been present or absent, but they heard the sound stimulus in three reverberant contexts: under anechoic conditions, with the addition of reverberation simulating a large room, and with reverberation simulating a small room.

3.10.2 Results

Figure 3.10 shows the average confidence ratings made by four listeners in the presence (circular symbols) and absence (diamond symbols) of a reflector, plotted as a function of reflector angle. A third-order polynomial was fit to the average ratings for each of these conditions, and the distance between these two curves gives an indication of how easily the sound of the reflector could be identified. Note that the highest ratings were given for an incidence angle of 45° (close to a specular reflection).

Figure 3.11 shows the average confidence ratings made in the presence and absence of an occluder, again plotted as a function of the angle of the obstruction, and using (circular symbols for occluder presence and diamond symbols for absence). However, in this case, the zero degree angle corresponds to the

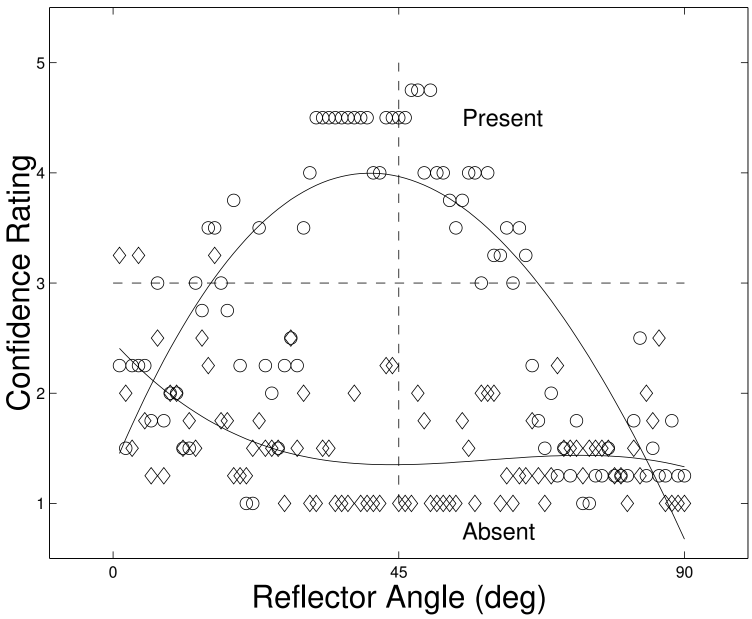


Figure 3.10: Identification of reflector presence

spatial configuration in which the maximum width of the occluder is interposed between source and sink. Thus, ratings were highest for low angles (i.e., greatest blocking). The smooth curves fit to these data converge only at the highest angles (near 90° incidence).

Figure 3.12 shows the relative frequency of each response for four listeners of the third experiment. With no evidence of a ceiling or floor effect, it appears that identification performance is not degraded by the presence of simulated reverberation.

3.10.3 Discussion

How can these data be used to determine which reflectors and occluders are most important to render? As stated before, the intention is to render only those that combine with the direct sound to provide an identifiably different spatial impression in comparison to the image of the direct sound alone. An arbitrary threshold confidence rating of "3" (marked by the horizontal dotted line shown in Figures 3.10 and 3.11) to choose which obstructions should be rendered might be used, but this approach does not take into account the number of rendering elements for which resources are available. An alternative is to choose the obstructions for rendering from a set containing both reflectors and occluders, using the standardized confidence rating scale as a

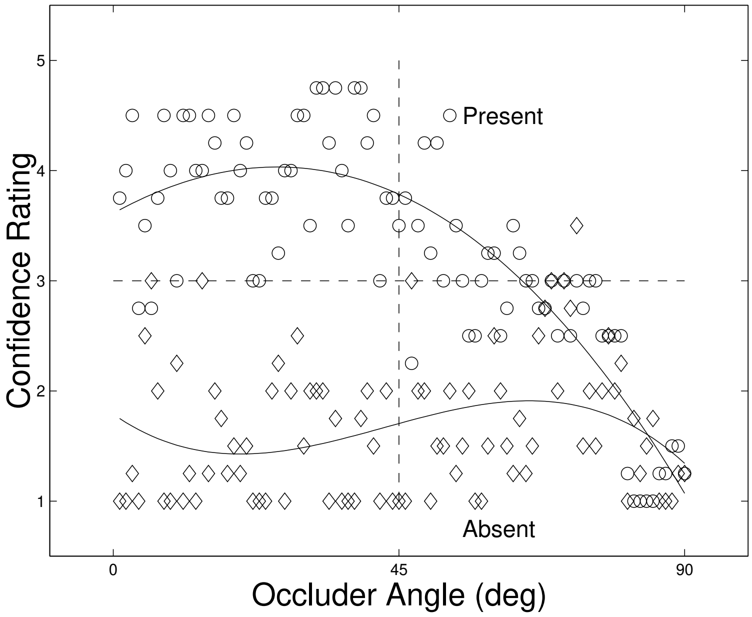


Figure 3.11: Identification of occluder presence

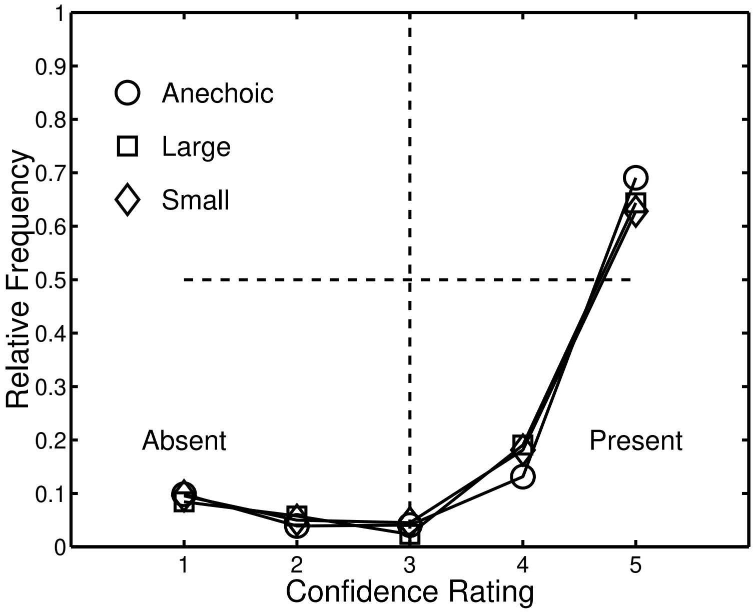


Figure 3.12: Identification of occluder presence in reverberation

common means for setting the criterion.

3.11 Sound spatialization with processing elements

A hardware architecture for sound spatialization proposed by [Ikedo and Martens, 1999] has processing elements for each sound path. The design includes obstruction transfer function. Such a system has processing elements as shown in Figure 3.13.

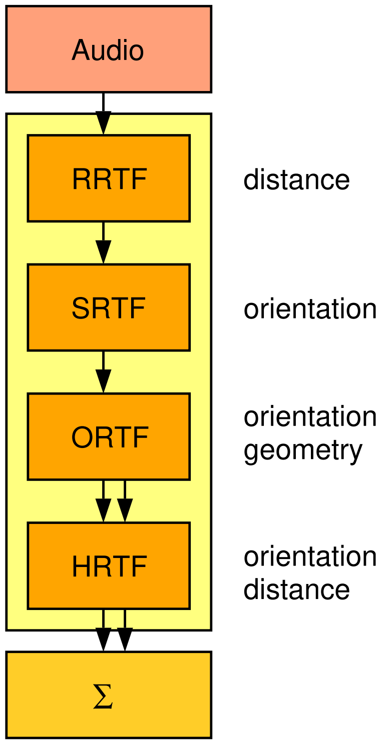


Figure 3.13: Processing element for one sound pass including control parameters

Range radiation transfer functions (RRTFs) model distance effects, parameterized by the distance between source and sink. Source radiation transfer functions model the radiation pattern of the sound source with orientation of the source relative to the sink as parameter. The hardware architecture can handle only a single obstruction. In case the obstruction is a reflection, then for each reflection, one processing element is used. The obstruction (occluder or reflector) transfer function (ORTF) has as parameters the geometry and orientation. The head-related transfer function (HRTF) uses orientation of the sink and distance as parameter. The responsibility of the resource manager is to allocate those processing elements and distribute control parameters.

3.12 Resource management for occluding objects

A processing element like that suggested in the previous section can handle only a limited number of occluders (i.e., one). A resource management system can select relevant occluders using Algorithm 3 and perceptual criteria investigated in Section 3.10.

Algorithm 3 Selecting occluder for sound sink path

```
for each sink in active sinks do
  for each source in active sources of sink do
    find all occluders along sink sound path
    calculate relevance based on perceptual criteria and geometry
    sort by relevance
    select most relevant occluders as resources are available
  end for
end for
```

Bibliography

[Anderson and Casey, 1997] David B. Anderson and Michael A. Casey. The sound dimension. *IEEE Spectrum*, 34(3):46–50, March 1997.

[Barron, 1971] M. F. E. Barron. The subjective effects of first reflections in concert halls — the need for lateral reflections. *Journal of Sound and Vibration*, 15:475–94, 1971.

[Begault, 1994] Durand R. Begault. *3-D Sound for Virtual Reality and Multimedia*. Academic Press, 1994. ISBN 0-12-084735-3.

[Begault, 1996] Durand R. Begault. Audible and inaudible early reflections: thresholds for auralization system design. In *Audio Engineering Society 100th Convention*, Copenhagen, 1996. Preprint 4244.

[Bell *et al.*, 1997] Gavin Bell, Rikk Carey, and Chris Marrin. ISO/IEC 14772-1:1997: The Virtual Reality Modeling Language (VRML97), 1997. http://www.vrml.org/Specifications/VRML97/.

[Blauert, 1968] Jens Blauert. Ein Beitrag zur Theorie des Vorwaerts - Rueckwaerts - Eindruckes beim Hoeren. In *The 6th International Congress on Acoustics*, pages A – 45–48, Tokyo, Japan, August 1968. (In German).

[Blauert, 1996] Jens Blauert. *Spatial Hearing: The Psychophysics of Human Sound Localization.* MIT Press, revised edition, 1996. ISBN 0-262-02413-6.

[Dhillon, 1994] Navdeep S. Dhillon. Cellular approach for modeling room acoustics: A framework for implementations based on the ray tracing algorithm. Master's thesis, University of Wisconsin - Madison, August 1994. http://home1.gte.net/dhillos/camra/index.html.

[Ellis, 1998] Sean Ellis. Towards More Realistic Sound in VRML. In *VRML 98*, pages 95–100, Monterey CA, USA, 1998.

[Funkhouser *et al.*, 1998] Thomas Funkhouser, Ingrid Carlbom, Gary Elko, Gopal Pingali, Mohan Sondhi, and Jim West. A beam tracing approach to acoustic modeling for interactive virtual environments. In *SIGGRAPH 98 conference*, held in Orlando, Florida, July 1998.

[Green and Swets, 1966] David M. Green and John A. Swets. *Signal Detection Theory and Psychophysics.* John Wiley & Sons, 1988 reprint edition, 1966. ISBN 0-932146-23-6.

[Handel, 1989] Stephen Handel. *Listening: An Introduction to the Perception of Auditory Events.* MIT Press, 1989. ISBN 0-262-08179-2.

[Herder and Cohen, 1997a] Jens Herder and Michael Cohen. Enhancing perspicuity of objects in virtual reality environments. In *CT'97 — Second Int. Cognitive Technology Conf.*, pages 228–237. IEEE, IEEE Press, August 1997. ISBN 0-8186-8084-9.

[Herder and Cohen, 1997b] Jens Herder and Michael Cohen. Sound Spatialization Resource Management in Virtual Reality Environments. In *ASVA'97 — Int. Symp. on Simulation, Visualization and Auralization for Acoustic Research and Education*, pages 407–414, Tokyo, Japan, April 1997. The Acoustical Society of Japan (ASJ).

[Ikedo and Martens, 1999] Tsuneo Ikedo and William L. Martens. Technologies for creating realistic sights and sounds in future multimedia systems. 1999. in preparation.

[Ishikawa *et al.*, 1998] Kimitaka Ishikawa, Minefumi Hirose, and Jens Herder. A sound spatialization server for a speaker array as an integrated part of a virtual environment. In IEEE *YUFORIC Germany 1998*, Stuttgart, June 1998. `http://www-ci.u-aizu.ac.jp/~herder/publications/ve98-spatial-server/`.

[Kendall and Martens, 1984] Gary S. Kendall and William L. Martens. Simulating the cues of spatial hearing in natural environments. In *ICMC: Proc.* **I***ntnl.* **C***omp.* **M***usic* **C***onf.*, pages 111–126, Paris, 1984. Computer Music Association.

[Kendall *et al.*, 1986] Gary S. Kendall, William L. Martens, Daniel J. Freed, M. Derek Ludwig, and Richard W. Karstens. Image model reverberation from recirculating delays. *Proceedings of the Audio Engineering Society 81st Convention*, 1986. Preprint No. 2408 (C-17).

[MacKenzie, 1995] I. Scott MacKenzie. Input devices and interaction techniques for advanced computing. In Woodrow Barfield and Thomas A. Furness III, editors, *Virtual Environments and Advanced Interface Design*, pages 436–470. Oxford University Press, 1995. ISBN 0-19-507555-2.

[Makous and Middlebrooks, 1990] James C. Makous and John C. Middlebrooks. Two-dimensional sound localization by human listeners. *JASA*, 87(5):2188–2200, May 1990.

[Martens and Herder, 1999] William L. Martens and Jens Herder. Perceptual criteria for eliminating reflectors and occluders from the rendering of environmental sound. In *Proc. Joint Meeting of the* 137^{th} *Regular Meeting of the Acoustical Society of America and the* 2^{nd} *Convention of the European Acoustics Association: Forum Acusticum*, page CDROM, Berlin, March 1999. Acoustical Society of America (ASA), and European Acoustics Association (EAA). Signal Processing in Acoustics and Psychological and Pysiological Acoustics: Auditory Displays, 1pSP2.

[Martens *et al.*, 1999] William L. Martens, Jens Herder, and Yoshiki Shiba. A filtering model for efficient rendering of the spatial image of an occluded virtual sound source. In *Proc. Joint Meeting of the* 137^{th} *Regular Meeting of the Acoustical Society of America and the* 2^{nd} *Convention of the European Acoustics Association: Forum Acusticum*, page CDROM, Berlin, March 1999. Acoustical Society of America (ASA), and European Acoustics Association (EAA). Signal Processing in Acoustics and Psychological and Pysiological Acoustics: Auditory Displays, 1pSP7.

[Mulgund and Zacharias, 1996] Sandeep S. Mulgund and Greg L. Zacharias. A situation-driven adaptive pilot/vehicle interface. In *Proceedings: 3rd Annual Symposium on Human Interaction with Complex Systems — HICS'96*, pages 193–198. IEEE, IEEE Computer Society Press, 1996. Dayton, August 25-28.

[Ovan and Havens, 1993] Russell Ovan and William S. Havens. Intelligent mediation: An architecture for the real-time allocation of interface resources. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, pages 55–61. ACM SIGCHI, ACM Press, 1993. Orlando, Florida, January 4-7, 1993.

[Sanders and McCormick, 1987] Mark S. Sanders and Ernest J. McCormick. *Human Factors in Engineering and Design.* McGraw-Hill, New York, sixth edition, 1987. ISBN 0-07-044903-1.

[Schiffman, 1995] Harvey Richard Schiffman. *Sensation and Perception: An Integrated Approach*, chapter 13, pages 350–363. John Wiley & Sons, Inc., 4th edition, 1995. ISBN 0-471-58620-X.

[Stille *et al.*, 1996] Stefan Stille, Shailey Minocha, and Rolf Ernst. A^2DL — An Adaptive Automatic Display Layout System. In *Proc. 3rd Annual Symposium on Human Interaction with Complex Systems — HICS'96*, pages 243–250. IEEE, IEEE Computer Society Press, 1996. Dayton, August 25-28.

[Sukaviriya and Foley, 1993] Piyawadee "Noi" Sukaviriya and James D. Foley. Supporting adaptive interfaces in a knowledge-based user interface invironment. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, pages 107–113. ACM SIGCHI, ACM Press, 1993. Orlando, Florida, January 4-7, 1993.

[Suzuki, 1999] Ikumi Suzuki. Elevation identification performance for real sound sources vs. HRTF-processed virtual sources. Bachelor thesis, University of Aizu, 1999.

[Takala and Hahn, 1992] Tapio Takala and James Hahn. Sound rendering. *Computer Graphics*, 26(2):211–220, July 1992.

[Tsingos and Gascuel, 1997] Nicolas Tsingos and Jean-Dominique Gascuel. Soundtracks for computer animation: Sound rendering in dynamic environments with occlusion. In *Graphics Interface '97*, pages 9–16, 1997.

[Tsingos and Gascuel, 1998] Nicolas Tsingos and Jean-Dominique Gascuel. Fast rendering of sound occlusion and diffraction effects for virtual acoustic environments. In *the 104th Convention of the AES*, Copenhagen, May 1998. Preprint 4699 (P4-7).

[Wenzel, 1994] Elizabeth M. Wenzel. Spatial sound and sonification. In Gregory Kramer, editor, *Auditory Display*, volume XVIII, pages 127–150. Addison-Wesley, April 1994. ISBN 0-0201-62603-9.

[Williams, 1994] Sheila M. Williams. Perceptual principles in sound grouping. In Gregory Kramer, editor, *Auditory Display*, volume XVIII, pages 95–125. Addison-Wesley, April 1994. ISBN 0-0201-62603-9.

Chapter 4

Optimization through Clustering

Level-of-detail is a concept well-known in computer graphics to reduce the number of rendered polygons. Depending on the distance to the subject (viewer), the objects' representation is changed. A similar concept is the clustering of sound sources, presented in this chapter. Clusters can be used to hierarchically organize mixels [Cohen, 1993, p. 294] and to optimize the use of resources, by grouping multiple sources together into a single representative source [Herder and Cohen, 1997]. Such a clustering process should minimize the error of position allocation of elements, perceived as angle and distance, and also differences between velocity relative to the sink (i.e., Doppler shift). Objects with similar direction of motion and speed (relative to sink) in the same acoustic resolution cone and with similar distance to a sink can be grouped together.

The basic idea of clustering is illustrated in Figure 4.1. Consider the cluster in the upper left corner. The flat ellipsoid surrounding a sound source represents the radiation pattern. The external vector denotes direction of motion and speed of the object. Imagine two cars on a road, chasing each other but not close to an observer. Both move away from the sound sink in the middle of the drawing. Similarly, the sources clustered in the upper right corner are not moving (imagine a group of people talking at a distance), and can be easily represented as a single source which mixes the signals of all sources in the cluster. The other sound sources cannot be clustered because they have different motion direction or do not fit into a single resolution cone (i.e., direction would be perceived differently).

The required information regarding velocity and moving direction is obtained via object monitoring, as described in Section 4.3. The sources in the lower part of the Figure 4.1 cannot be clustered because of different motion direction (i.e., different Doppler shift).

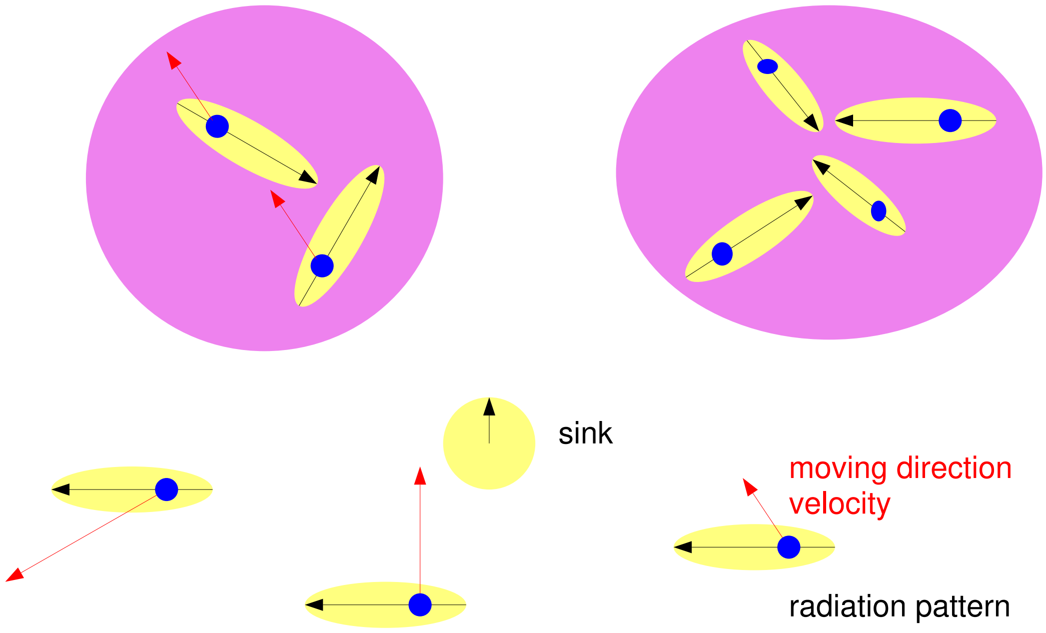


Figure 4.1: Clustering of sources in resolution cone with similar moving direction and speed: the cluster in the left upper corner shows two cars chasing each other in the distance in direction away from the sink; the cluster in the right upper corner represents a stationary group of people talking; the other sound sources cannot be clustered because of different motion direction, or because they do not fit into one resolution cone

4.1 Clustering algorithm

The sound resource allocation algorithm described in Chapter 2 can be extended and improved by introducing sound source clustering. Figure 4.2 shows how clustering is included into the algorithm. The previous algorithm is used for calculating the set of audible sources, but does not evaluate the priorities before clustering takes place. After clustering, priorities can be used to determine the set of active source for audio rendering.

Clustering Algorithm 4 is presented in pseudo-code. A sound source is added to a cluster if the perceptual error between representative (i.e., virtual) sound source and all sound sources in the cluster is smaller than an experimentally determined threshold (e.g., using data obtained by [Makous

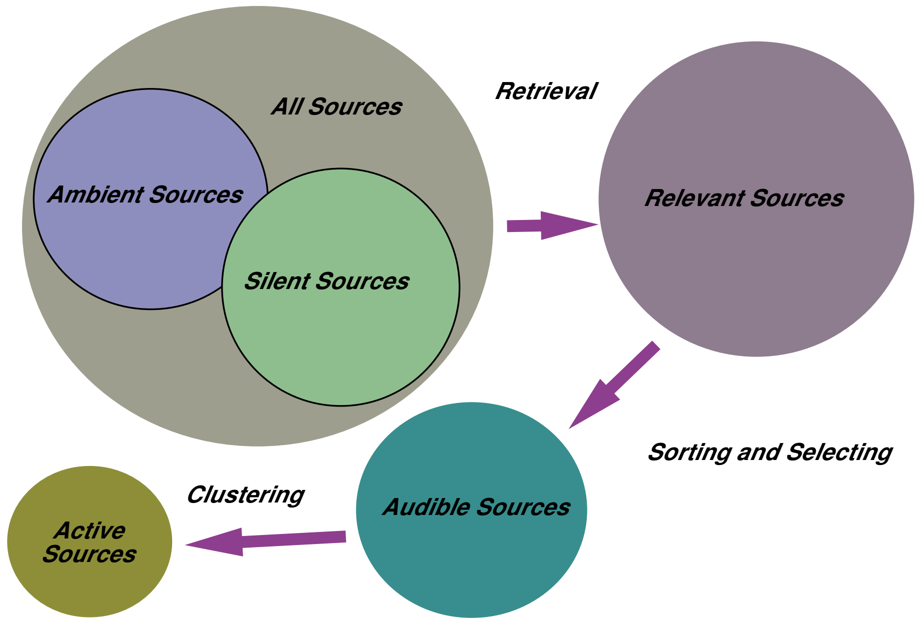


Figure 4.2: Clustering reduces the number of required spatialization channels

and Middlebrooks, 1990]). Error can be calculated for direction, distance, and Doppler shift. A cluster is valid for only one sink. Figure 4.3 shows an example in which two sound sources are clustered together and represented by a representative sound source. The sound sources are within the resolution cone of the representative sound source. The resolution cone shape varies depending on azimuth and elevation.

Clustering Algorithm 4 converges quickly, because in the while loop, the `workSet` is reduced in the worst case at least by one source. (The number of steps for the while loop is $\sum_{i=1}^{n} n - i = 1/2n(n+1)$.) The complexity is $O(n^2 * m)$, where n is the number of sound sources and m is the number of sound sinks. The algorithm is not optimal in the sense that there might be another clustering configuration which has fewer clusters. An optimal algorithm would calculate all possible configurations and would choose that configuration with the fewest clusters. minimize perceptual errors as well as number of clusters

The `representative` function returns an aggregate sound source for a set of sound sources. The position of that representative source can be the centroid (mean position) of the set, or calculated as suggested in the next section.

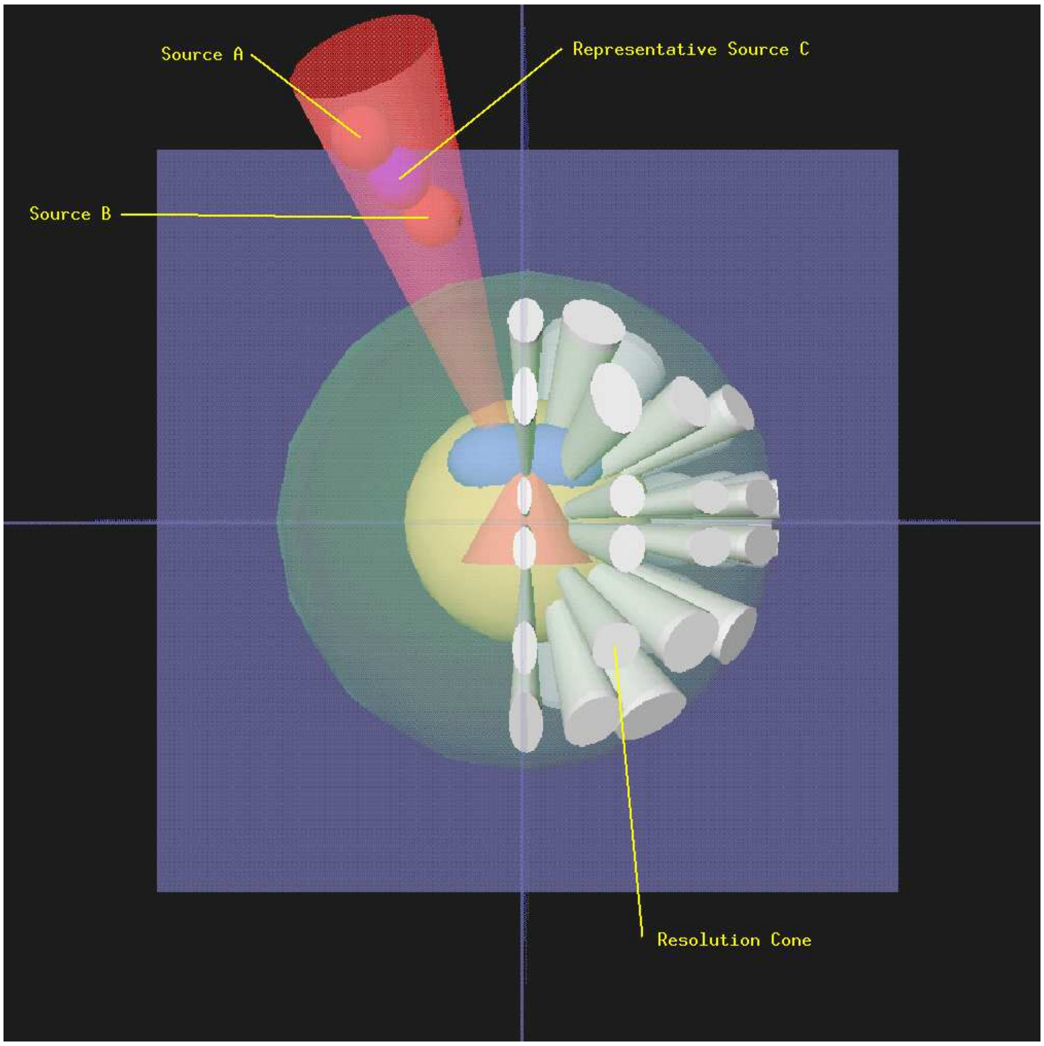


Figure 4.3: Two sound sources A and B are clustered together within the resolution cone of the representative virtual sound source C

Algorithm 4 Clustering algorithm for sound sources

```
for each sink in sinks do
  workSet ⇐ sources of sink
  while workSet is not empty do
    add source from workSet to representativeSourceSet
    remove source from workSet
    for each source in workSet do
      representativeSource ⇐
          representative(representativeSourceSet + source)
      if withinNotPerceivableLimits(sink,representativeSource,
                                    representativeSourceSet + source)
      then
        add source to representativeSourceSet
        remove source from workSet
      end if
    end for
    add representative(representativeSourceSet) to representativeSources
    including mixing data
  end while
end for
```

$$\texttt{representative}(sources) \;=\; \frac{1}{n}\sum_{i=1}^{n} source_i \qquad (4.1)$$

The boolean `withinNotPerceivableLimits` function returns `true` if the sources are within the spatial not perceivable limits (i.e., localization errors).

$$\begin{aligned}&\texttt{withinNotPerceivableLimits}(sink, representative, sources) = \\ &\quad \texttt{withinNotPerceivableDirection}(sink, representative, sources)\ \& \\ &\quad \texttt{withinNotPerceivableDistance}(sink, representative, sources)\ \& \\ &\quad \texttt{withinNotPerceivableDoppler}(sink, representative, sources) \quad (4.2)\end{aligned}$$

The boolean `withinNotPerceivableDirection` function returns `true` if the sources are in the resolution cone of the representative for a given sink. The azimuth and elevation limit values for the specific direction of the representative are calculated by interpolation of experimentally determined limit values.

The boolean `withinNotPerceivableDistance` function returns `true` if sources are in the range limits of the representative for a given sink. The range limit values for a specific distance are calculated by interpolation of experimentally determined limit values.

The boolean `withinNotPerceivableDoppler` function compares the Doppler shift of all sources relative to the Doppler shift of the representative. If the difference in Doppler shift is not perceivable, then the function returns `true`. Again the limit values are based on experimentally determined limit values.

4.2 Determining a representative sound source location for a cluster

A cluster of sound sources can be represented by one (representative) source which is then passed to a spatialization backend. A straightforward approach is to calculate the first moment of all sources in the cluster [Suzuki, 1997]. This does not consider the shape of the perceptual space. Section 3.8 explains the cone of confusion, in which sound sources on the rings centric to the binaural axis cannot be distinguished just by interaural time delay and interaural intensity difference. Usual studies of localization errors use the polar coordinate system using azimuth and elevation. If localization errors are presented using the coordinate system by [Morimoto and Aokata, 1984] based on a lateral and rising angle, then accuracy can be explained by two mutually independent cues. A similar coordinate system is suggested in this section.

A representative virtual source for two sound sources on such a ring should be also on the ring. Taking this requirement into account suggests using a cylindrical coordinate system, as shown on the left of Figure 4.5. A location *loc* (see Figure 4.4) is represented as a triple consisting of the distance y along the interaural axis, the length r of an orthogonal vector to y from the interaural axis to the location of the source, and the angle of this vector relative to the line-of-sight vector.

$$loc = \begin{pmatrix} y \\ r \\ \varphi \end{pmatrix} \tag{4.3}$$

In general, a representative source location is defined as

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.4}$$

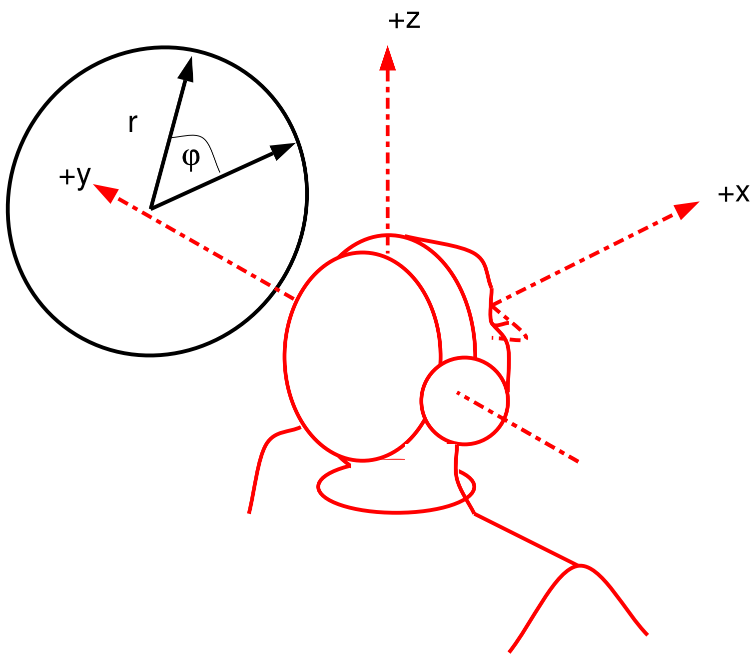


Figure 4.4: Listener inside the cylindrical coordinate system

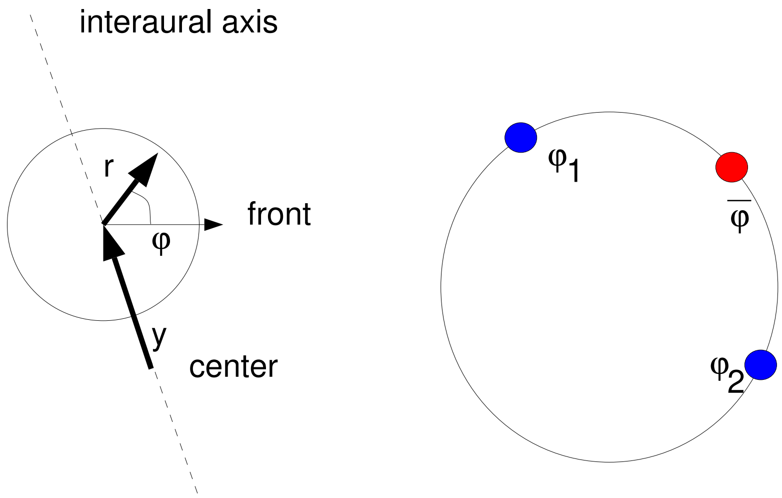


Figure 4.5: left: Cylindrical coordinate system; right: Representative source location

$$\overline{r} \quad = \quad \frac{1}{n}\sum_{i=1}^{n} r_i \tag{4.5}$$

To determine the angle $\overline{\varphi}$, several cases must be distinguished. Before that, some definitions are introduced.

Let

$$x_i = (\cos(\varphi_i), \sin(\varphi_i)) \tag{4.6}$$

The vector x_i for a given angle φ_i is on the unit circle. The first moment of the x_i corresponding to the angle $\overline{\varphi}$ is used to resolve the ambiguity and to find the minimal dividing angle. In the case that two sound sources are on the same ring, then $y_1 = y_2 = \overline{y}$ and $r_1 = r_2 = \overline{r}$ and the representative source is on the same ring.

The first momment vector $\overline{x}$ is defined as follows:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.7}$$

Let $\tilde{x}$ be the projection of $\overline{x}$ to the unit circle.

Set by definition:

$$\alpha = \arctan(\frac{\sum_{i=1}^{n}\sin\varphi_i}{\sum_{i=1}^{n}\cos\varphi_i}) \in (-\frac{\pi}{2}, \frac{\pi}{2}) \tag{4.8}$$

Case 1 $\overline{x} = 0$:
Set $\overline{\varphi} = 0$, i.e., use (1,0) $= \tilde{x}$

Case 2 $\overline{x} \neq 0$:
$\overline{x} = \frac{1}{n}(\sum_{i=1}^{n} \cos\varphi_i, \sum_{i=1}^{n} \sin\varphi_i)$

Case 2.1 $\sum_{i=1}^{n} \cos\varphi_i = 0$:

Case 2.1.1 $\sum_{i=1}^{n} \sin\varphi_i > 0$:
Set $\overline{\varphi} = \frac{\pi}{2}$, i.e., use $(0, 1) = \tilde{x}$

Case 2.1.2 $\sum_{i=1}^{n} \sin\varphi_i < 0$:
Set $\overline{\varphi} = \frac{3\pi}{2}$, i.e., use $(0, -1) = \tilde{x}$

Case 2.2 $\sum_{i=1}^{n} \cos\varphi_i \neq 0$:

Case 2.2.1 $\overline{x}_1 > 0, \overline{x}_2 \geq 0$:
$\overline{\varphi} = \alpha \in [0, \frac{\pi}{2})$

Case 2.2.2 $\overline{x}_1 < 0$:
$\overline{\varphi} = \alpha + \pi \in (\frac{\pi}{2}, \frac{3\pi}{2}), \alpha \in (-\frac{\pi}{2}, \frac{\pi}{2})$

Case 2.2.3 $\overline{x}_1 > 0, \overline{x}_2 < 0$:
$\overline{\varphi} = \alpha + 2\pi \in (-\frac{3\pi}{2}, 2\pi), \alpha \in (-\frac{\pi}{2}, 0)$

In the case the sound sources are in the horizontal plane (i.e., φ for all sources is 0), then the representative source location is the first central moment of the vectors in this plane as a straightforward calculation shows.

4.3 Object monitoring

Object monitoring is necessary for two reasons. For a clustering heuristic, information about speed and motion direction is necessary. If not considered, the Doppler effect would ignored. In other words, objects with different Doppler shifts should not be combined into a cluster. Certain kinds of sound production are atomic, which means they cannot be decomposed without getting a different perception (e.g., attack portion of hitting a string). This situation occurs, e.g., using the MIDI protocol between "`note on`" and "`note off`" commands.

Taking this into account, the number of resource assignment (to cluster, to channel, to ambient, ...) switches should be minimized. Object monitoring allows look-ahead and resource reservation. The following object attributes can be monitored or calculated:

- speed;
- moving direction;
- changes of speed;
- changes of moving direction;
- probability of attribute change;
- maximum, minimum, and average values[1] for direction;
- intensity;
- location; and
- priority.

The probability of an attribute change can be determined by counting attribute changes and dividing through the passed time.

4.4 Discussion

The advantages and disadvantages of clustering can be summarized as:

- far better use of spatialization resources,
- freeing resources for other tasks such as visualization,
- improved spatialization fidelity in case of limited resources,
- perceptual artifacts might occur during switching (source assignment to different cluster),
- costs for mixing the audio streams might reduce the gains through clustering, and
- sound spatialization errors might occur through averaging object attributes.

[1] see Application Programmer Interface in Chapter 5

Bibliography

[Cohen, 1993] Michael Cohen. Throwing, pitching, and catching sound: Audio windowing models and modes. *IJMMS: the Journal of Person-Computer Interaction*, 39(2):269–304, August 1993. ISSN 0020-7373.

[Herder and Cohen, 1997] Jens Herder and Michael Cohen. Sound Spatialization Resource Management in Virtual Reality Environments. In *ASVA'97 — Int. Symp. on Simulation, Visualization and Auralization for Acoustic Research and Education*, pages 407–414, Tokyo, Japan, April 1997. The Acoustical Society of Japan (ASJ).

[Makous and Middlebrooks, 1990] James C. Makous and John C. Middlebrooks. Two-dimensional sound localization by human listeners. *JASA*, 87(5):2188–2200, May 1990.

[Morimoto and Aokata, 1984] Masayuki Morimoto and Hitoshi Aokata. Localization cues of sound sources in the upper hemisphere. **J**. **A***cous.* **S***oc.* **J***ap.*, 5(3):165–173, 1984.

[Suzuki, 1997] Taku Suzuki. Spatialization resource management for MIDI mixels. Bachelor thesis, University of Aizu, 1997.

Chapter 5

Sound Spatialization Application Programmer Interface

For the application programmer interface, the VRML97 standard [Bell *et al.*, 1997] [Bell *et al.*, 1996] [Carey and Bell, 1997] [Hartman and Wernecke, 1996] was chosen.

The VRML97 specification defines a file format and semantic interpretation. For sound support, only two nodes, `Sound` and `AudioClip`, are specified. It is assumed that the sink (i.e., listener) is at camera position and/or controlled by the viewer.

The VRML97 specification was extended by this research for nodes specified in this chapter to enable more sophisticated audio modeling and rendering. The nodes are

`SfSoundSink` sound sink, an independent receiver (e.g., dummy head, microphone),

`SfSoundScape` soundscape, scope limitation and room acoustics,

5.1 Audio rendering process

A scenegraph defines a scene with graphical, interactive, acoustical (Figure 5.1), and behavior nodes and can be constructed either by class instantiation or external files which can be created using other authoring tools. Each node is defined in its own local coordinate system. During an audio rendering pass all transformations are resolved and necessary audio control data passed to the resource manager. Resource management and final rendering in a spatialization backend are calculated in world coordinates. Resource management involves mapping from source⟶sink channels to available mixels, spatialization channels, including a scheme to predict the perceptual relevance of a sound source in a given configuration. Resources are used economically by applying a clustering technique which mixes spatially proximate sound sources, representing them as a single sound (representative) source.

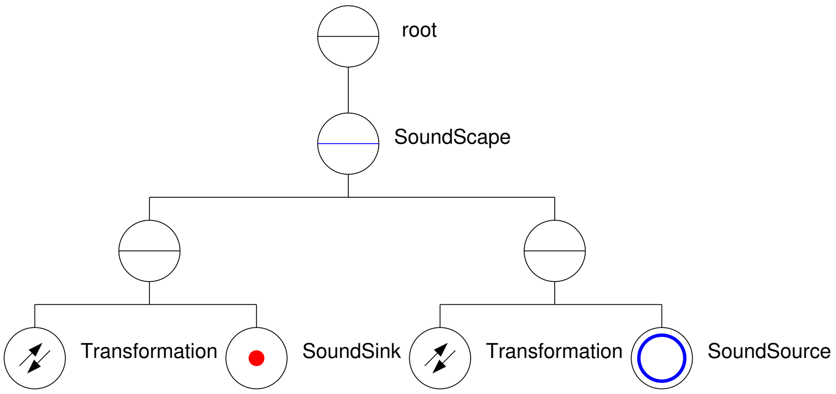


Figure 5.1: Scenegraph with sound objects

5.2 Sound source node

```
SfSoundSource {
  exposedField SFVec3f  direction     0 0 1
  exposedField SFFloat  intensity     1
  exposedField SFVec3f  location      0 0 0
  exposedField SFFloat  maxBack       10
  exposedField SFFloat  maxFront      10
  exposedField SFFloat  minBack       1
  exposedField SFFloat  minFront      1
  exposedField SFFloat  priority      0
  exposedField SFNode   source        NULL
  field        SFBool   spatialize    TRUE
}
```

Figure 5.2: Sound source node specification

Figure 5.2 shows the specification of a sound source. `Location` and `direction` are defined within the object space. The `intensity` inside of the core range ellipsoid, which is given by `minBack` and `minFront`, is one.

Fields in a sound source node:

direction specifies a primary sound-emmision direction as "front," and specified as vector defining the major axis of the audible-sound ellipsoids.

intensity adjusts the gain value of the sound source. An intensity of 0 denotes silence, and an intensity of 1 is the full gain as provided by the AudioClip node.

location specifies the position of the sound source in object space.

maxBack is the distance in the direction opposite the `direction` vector to which the audible range ellipsoid extends.

maxFront is the distance along the `direction` vector to which the audible range ellipsoid extends.

minBack is the distance in the direction opposite the `direction` vector to which the core ellipsoid extends.

minFront is the distance along the `direction` vector to which the core range ellipsoid extends.

priority is a hint to the sound spatialization resource management about how important the sound is. It should be left at 0 for background sounds, and set to 1 to ensure the display of important, short single-event sounds.

source is an AudioClip node which specifies how the sound will be generated (not spatialized); if not specified, the Sound node emits no sound.

spatialize indicates whether the sound should be played as if it's at a particular point in space (TRUE), or whether it should be rendered as ambient background sound (FALSE).

Taking a geometrical model into account, the resource management can calculate the volume at listener position from the `direction` and `location` fields. The `spatialize` and `priority` fields are used besides other indicators like loudness and scope for selecting sources which will get a channel assignment.

5.3 Sound sink node

Exocentric views and multiple sinks motivated the extension to the VRML97 standard with a node which represents a sound sink, similar to a camera description. The sound sink node shown in Figure 5.3 allows separate control of listening location and viewpoint and forms the basis of a general scheme for multiple sinks (see Section 2.2.3).

```
SfSoundSink {
  exposedField SFRotation  orientation   0 0 1 0
  exposedField SFFloat     sensitivity   1
  exposedField SFVec3f     location      0 0 0
  exposedField SFFloat     farDistance   10
  exposedField SFFloat     nearDistance  1
  exposedField SFFloat     priority      0
  field        SFBool      enable        TRUE
}
```

Figure 5.3: Sound sink node specification

Fields of a sound sink node:

orientation is defined as a rotation of the sound sink direction from its default (0,0,-1) vector. The up direction is (0,1,0). This field, along with the current geometric transformation, specifies the orientation of the sound sink in world coordinates.

sensitivity adjusts the gain of the incoming sound signals; an intensity of 0 indicates total deafness, and an intensity of 1 indicates full gain.

location is the position of the sound sink in object space.

farDistance is the radius of the sensible range sphere.

nearDistance is the radius of the core sphere (inner ear distance in case of HRTF-based processing [CRE, 1994]).

priority is a hint to the sound spatialization resource management about how important this sound sink is, especially applicable in case of multiple sinks.

enable indicates whether the sink should contribute to the spatialization (applicable in case of multiple sinks).

5.4 Soundscape node

Inspired by the Java3D specification [Sowizral *et al.*, 1997], the soundscape nodes (Figure 5.4 relates medium definition and space. The bounding box limits the range of aural attributes specifications.

```
SfSoundScape {
  exposedField SoSFVec3f min                        0 0 0
  exposedField SoSFVec3f max                        1 1 1
  exposedField SoSFAuralAttributes attributes  NULL
}
```

Figure 5.4: Soundscape node specification

Fields of a soundscape node:

min is the first corner of the bounding box.

max is the second corner of the bounding box.

attributes SfAuralAttributes node specifing the aural attributes.

Bibliography

[Bell *et al.*, 1996] Gavin Bell, Rikk Carey, and Chris Marrin. The Virtual Reality Modeling Language, Version 2.0 Specification, ISO/IEC CD 14772, August 1996. http://www.vrml.org/VRML2.0.old/.

[Bell *et al.*, 1997] Gavin Bell, Rikk Carey, and Chris Marrin. ISO/IEC 14772-1:1997: The Virtual Reality Modeling Language (VRML97), 1997. http://www.vrml.org/Specifications/VRML97/.

[Carey and Bell, 1997] Rick Carey and Gavin Bell. *The Annotated* VRML *2.0 Reference Manual*. Addison-Wesley Developers Press, 1997. ISBN 0-201-41974-2.

[CRE, 1994] Crystal River Engineering, Inc. *CRE_TRON Library Reference Manual*, August 1994. Revision B.

[Hartman and Wernecke, 1996] Jed Hartman and Josie Wernecke. *The* VRML *2.0 Handbook*. Addison-Wesley, Inc., 1996.

[Sowizral *et al.*, 1997] Henry Sowizral, Kevin Rushforth, Michael Deering, Warren Dale, and Daniel Petersen. Java^{TM} 3D API Specification. Sun Microsystems, August 1997. `http://www.javasoft.com/products/java-media/3D/forDevelopers/3Dguide/j3dTOC.doc.html`.

Chapter 6

Abstract Spatialization Backend Interface

The abstract spatialization backend interface can be thought of as an abstraction layer between spatialization resource management and spatialization backends. This concept is also known as a hardware abstraction layer, but the backend is not necessary a piece of hardware, so that we do not want to use this term.

The spatialization device interface is an abstract interface to spatialization devices (e.g., Acoustetron and PSFC [Amano *et al.*, 1996]). This interface ensures that the resource manager (and an application) does not need to be changed if a new device is to be supported. Only a new spatialization device driver, derived from a template, needs to be developed.

The interface informs the resource manager about number of spatialization channels, number of non-spatialization channels, spatialization channel identifiers, non-spatialization channels, Doppler shift support, and volume spatial resolution (e.g., minimum audible angles).

In the other direction the resource manager informs the spatialization device about soundscape and system changes.

A second abstract interface exists for audio channels. A channel can get audio data from a MIDI synthesizer, a sound file player, or a port (e.g., mic). If available, a channel (player device) returns to the spatialization resource manager via the interface frequency and volume in normalized form (i.e., float between 0. and 1.). If a source is set inactive by the resource manager, then the related channel can also be stopped.

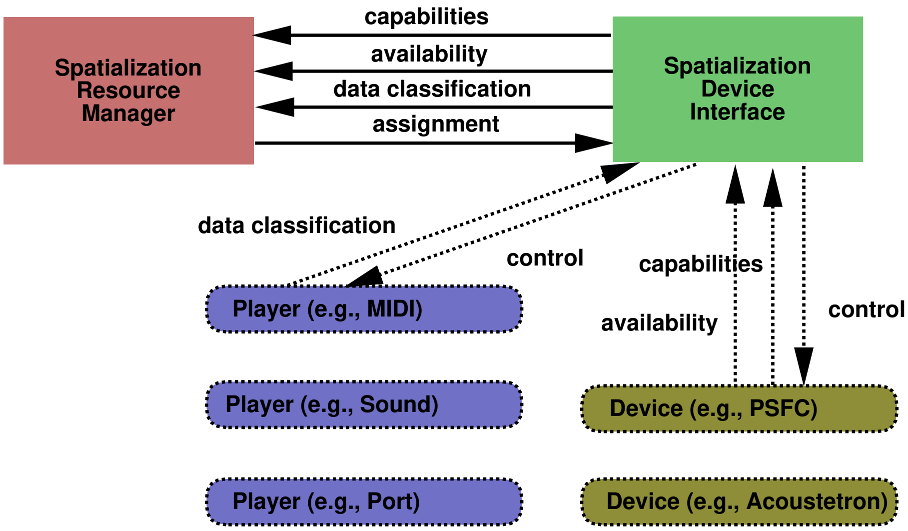


Figure 6.1: Spatialization device interface

6.1 Sound spatialization backends

The developed resource management was tested with different spatialization backends. Backends differ in their way to spatialize sound, the provided interface and realism. The developed algorithm for spatial sound resource management tries to cover all kind of spatialization backends. As representatives for different classes of spatialization backends the following sections introduce the PSFC, a loudspeaker array and the Acoustetron II, a HRTF-based system. The common features are shown and a method to unify the interfaces for the resource management.

6.1.1 Pioneer Sound Field Controller

The PSFC, or **P**ioneer **S**ound **F**ield **C**ontroller [Amano *et al.*, 1996] [Amano *et al.*, 1998], is a DSP-driven hemispherical loudspeaker array, installed in the Synthetic World Zone at the University of Aizu Multimedia Center. The PSFC system features realtime configuration of an entire sound field, including sound direction, virtual distance, and simulated environment (virtual room characteristics: reverberation level, room size and liveness) for each of two sources. It can also configure a dry (DSP-less) switching matrix for direct directionalization. The PSFC speaker dome is about 10 m in diameter, accommodating about fifty simultaneous users and allowing about twenty users at once to comfortably stand or sit near its sweet spot. Collocated with a large screen rear-projection stereographic display, the PSFC is intended for advanced multimedia and **v**irtual **r**eality applications. The screen and some of the speakers are shown in Figure 6.2.

The PSFC can configure both the context and the content of a virtual sound field: the context is the ambiance or presence — the room size, liveness, reflection pattern, and overall level; the content is the source direction — azimuth, elevation, and suggested distance. The direct sound moves by amplitude panning; the reflected sound moves by rotating the impulse response; the reverberant sound is orientation-independent.

The hemispherical speaker array's audio presentation is complemented by the wide-screen visual presentation. The spatially immersive environment provides a natural group experience, wide visual field-of-view, comfort (no fatigue-inducing or cumbersome head-mounted display). Such a system can be described as "roomware," software for a room, putting the users *inside* the computer [Brooks, 1997]. This notion is also related to the idea of an "immobot," an immobile robot that concentrates on attending and servicing the needs of collocated human users, rather than the traditional robotic tasks of exploration and manipulation of an external environment.

The PSFC's omnidirectional sound field complements the exaggerated visual display's inadequacies for immersion.

Direct sound directionalization

Source direction is set by programming azimuth θ and elevation ϕ for each of two independent source channels, which can move around with an update cycle of less than 100 ms. The transfer functions for are interpolated at the audio sampling rate (and not at the slower control rate), avoiding reconstruction (aliasing) problems. The signal is reproduced by three loudspeakers surrounding the sound image, the level of each loudspeaker determined by the distance between the projected image source and each loudspeaker.

Early reflection

Early reflection patterns were calculated via an image source model from architectural drawings, modeling walls, ceiling and floor with characteristic reflection coefficients to obtain intensity, delay, and direction parameters [Meyer *et al.*, 1965] [Kendall and Martens, 1988]. The FIR filters, room-related impulse responses, were generated by simulating transmission through the solid angle subtended by the respective speakers. Rather than a data-intensive, time-domain FIR filter (specifying amplitudes at a sampling rate) or a frequency-domain FIR filter (multiplying the FFTs of the sources

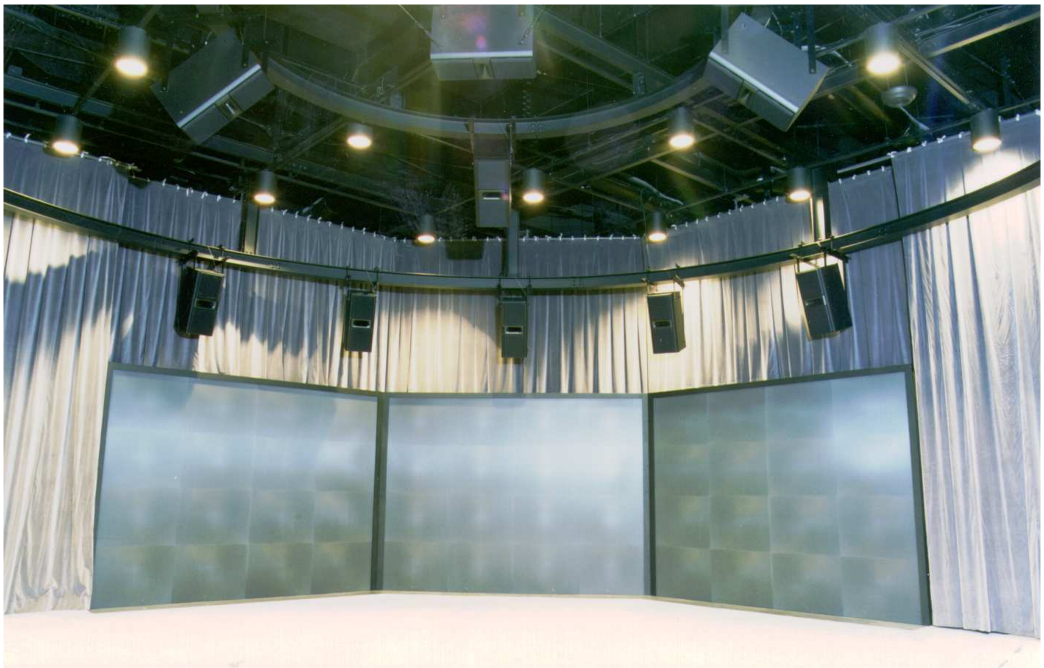


Figure 6.2: Multimedia Center: Virtual Reality Zone

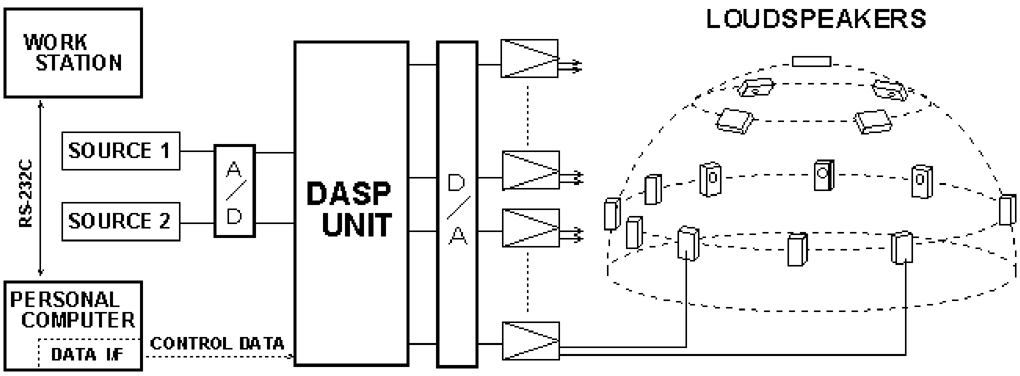


Figure 6.3: Multimedia Center: Pioneer Sound Field Controller

and transfer functions), early reflection modeling is implemented as a table of discrete (potentially sparse) delay time/normalized level pairs, the reflection being generated via a delay network. As implemented in the PSFC, early reflections are series of pure impulses, instead of the results of convolution with a realistic wall response. To simplify the taps, reflection patterns were elided (pruned) whenever the simulated amplitude fell below a certain threshold or when the reflection closely followed another stronger (masking) one. For each of the two source signals, up to seven reflections are reproduced from each of the twelve loudspeakers, for a total of 168. To correct the greater distance between the lower ring of speakers and the center of the room compared to the upper ring, delay is added to the signals bound for upper speakers.

Presence and ambiance: Room size, liveness, and reverberation The perceived sound quality of the simulated sound field is an interaction of programmed room size, liveness, and reverberation parameters.

Room size commands contribute to the perceived auditory spaciousness of the simulated environment. The apparent distance of reflective walls can be changed by varying the initial time gap (a.k.a. predelay) separating the arrival time of the direct sound from that of the first early reflection between 0–500 ms for each source [Tohyama *et al.*, 1995].

Overall volume is controlled by level, but control of the simulated environment's liveness (i.e., how reflective the walls, floor, and ceiling are) is accomplished by scaling the decay rate of simulated early reflections. The room liveness parameter is a kind of time constant, which contributes to the perceptual response characteristic termed "definition" [Rasch and Plomp, 1984]. Definition can be predicted from the ratio of the sound energy (sum of squares of coefficients) of the first 50 ms of the impulse response (including direct sound) to the total sound energy of the impulse response. As illustrated by Figure 6.4, the liveness also implicitly determines the level of the following reverberation, since the reverberation level of each liveness value was prescaled to ensure smooth continuity. Changes in room size and liveness can be effected within a latency of 300 ms.

The reverberation level parameter contributes both to how spacious the simulated environment seems, and indirectly, to how distant the sources seems. As discussed above, the reverberation level is scaled down if the liveness is reduced. But for a given liveness, the apparent distance of the sound source depends on the relationship between the level of the direct sound and the level of indirect sound. For a given virtual room simulation, varying the gain on a source relative to a fixed level of reverberant energy provides a strong cue to source distance. The distance so-cued is called the

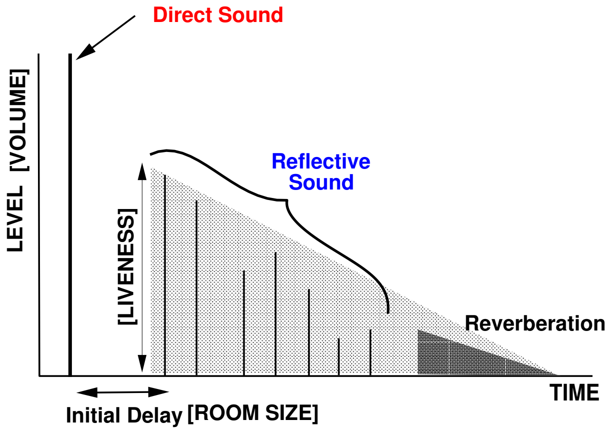


Figure 6.4: Early reflections and reverberation

"indirect-to-direct ratio" (IDR). In the PSFC, the IDR is controlled by simultaneously scaling the output level (which includes direct, reflective, and reverberant sound) and inversely adjusting the liveness (which includes only indirect sound). For example, low gain and high liveness suggest a distant source, while high gain and low liveness suggest a close source.

Exploiting the limited resolution of human hearing, the reverberation patterns are exponentially decayed noise filters, representing a Poisson distribution with time-increasing average (decreasing amplitude and increasing density), independent of source position.

6.1.2 Acoustetron II: an HRTF-based system

The Acoustetron II, developed by Chrystal River Engineering, is based on research at NASA Ames Research center [Foster *et al.*, 1991] [Begault, 1994, p. 205–208] and has its own API [CRE, 1994]. Sound sources are spatialized using head-related transfer functions for headphones. The HRTFs in the system are measured for a constant distance. The number of available spatialization channels vary depending on number of installed convolution boards (a typical configuration has 8 channels.) The system does not provide reverberation for distance cues or room impression. First-order reflection can be configured for "shoe box" (rectangular prism) environments, where each reflection takes up one spatialization channel. Doppler shift is supported.

6.1.3 MIDI: simple spatialization

Simple spatialization can be performed within a MIDI synthesizer or tone generator. Stereo panning and intensity control can give approximate spatialization adequate for many applications. More advanced spatialization would use pitch shift to implement the Doppler effect and mimicking interaural time delays. Control over reverberation would allow better distance cues and room effects. Further, occluder effects can be implemented through simple filtering. Experiments of using pitch shift for interaural time delays using two MIDI channels show problems regarding unpredictable time delays in the execution of pitch shift commands [Ishikawa, 1999]. Implementing MIDI spatialization on the level of a synthesizer might solve those problems.

6.2 Specification

Table 6.1 lists abstract interface functions for spatialization backends. Not all functions are listed, but only those related to the resource management process. Details are documented in the manual of the Sound Spatialization Framework [Herder, 1998].

Besides initialization and termination of a connection to a spatialization backend, primary functions are for updating source and sink data. The framerate for this updating process can be limited not to overburden the spatialization backend. Listener movements are passed via the `updateSink` function using sink location and orientation as parameters. Sound source changes like location, direction, and intensity are passed by the function `updateSource`.

6.3 Comparing the backend interfaces and unification

Table 6.2 shows which interface functions are used to implement the abstract interface for several backends.

Neither the PSFC nor the MIDI backend have functions for sink positioning, but the functionality can be achieved by translation of all sources.

6.4 Discussion of the features

Table 6.3 lists the different features of spatialization backends used within the prototype of the framework. The table does not show the quality of

function	parameters	description
`init`	`numberOfHeads`	Initialize the spatialization module and return the number of available mixels (spatialization channels).
`close`	`fromHead`, `toHead`	Shutdown the spatialization module. This can be limited to a range of of sinks using the parameters `fromHead` and `toHead`.
`updateSource`	`sourceIndex`, `location`, `direction`, `intensity`	Update a sound source. If the `sourceIndex` doesn't exist yet, create a new sound source. The source gain level is given using the parameter `intensity`, which is a normalized (0 ...1).
`updateSink`	`sinkIndex`, `location`, `orientation`	Update a sound sink. If the `sinkIndex` dosn't exist, a new sound sink is created. The location is given in world coordinates.
`removeSink`	`sinkIndex`	Removes a sound sink from the registered sinks.

Table 6.1: Abstract spatialization backend interface

function	PSFC	MIDI	Acoustetron II
initialize	`CL_PSFCinit`	`MD_RESETALL-CONTROLLERS`	`cre_init`
update source	`CL_Coordinates`	`MD_PAN` `MD_CHANNELVOLUME`	`cre_locate_source` `cre_amplify_source`
update sink	`CL_Coordinates`	`MD_PAN` `MD_CHANNELVOLUME`	`cre_locate_head` `cre_update_audio`

Table 6.2: Comparison between spatialization backend interfaces

immersion which can be achieved with different backends.

feature	PSFC	MIDI	Acoustetron II
Doppler effect	no	no	yes
mixels	2	16-32	8/12
output channels	14	2	2
distance cue	yes	yes	yes
left/right localization	yes	yes	yes
front/back localization	yes	no	yes
up/down localization	yes	no	yes
participants	20	limited by audio system	1

Table 6.3: Comparison between spatialization backend features

Bibliography

[Amano *et al.*, 1996] Katsumi Amano, Fumio Matsushita, Hirofumi Yanagawa, Michael Cohen, Jens Herder, Yoshiharu Koba, and Mikio Tohyama. PSFC: the Pioneer Sound Field Control System at the University of Aizu Multimedia Center. In RO-MAN*'96 - 5th* IEEE *International Workshop on Robot and Human Communication*. IEEE, November 1996.

[Amano *et al.*, 1998] Katsumi Amano, Fumio Matsushita, Hirofumi Yanagawa, Michael Cohen, Jens Herder, William Martens, Yoshiharu Koba, and Mikio Tohyama. A Virtual Reality Sound System Using Room-Related Transfer Functions Delivered Through a Multispeaker Array: the PSFC at the University of Aizu Multimedia Center. *TVRSJ:* **Trans**. *of the* **V***irtual* **R***eality* **S***ociety of* **J***apan*, 3(1):1–12, March 1998. ISSN 1342-4386.

[Begault, 1994] Durand R. Begault. *3-D Sound for Virtual Reality and Multimedia*. Academic Press, 1994. ISBN 0-12-084735-3.

[Brooks, 1997] Rodney A. Brooks. The Intelligent Room Project. In Jonathon P. Marsh, Chrystopher L. Nehaniv, and Barbara Gorayska, editors, *CT'97 - Second International Cognitive Technology Conference*, pages 271–278. IEEE, IEEE press, August 1997. Aizu-Wakamatsu, Japan, August 25-28, 1997.

[CRE, 1994] Crystal River Engineering, Inc. *CRE_TRON Library Reference Manual*, August 1994. Revision B.

[Foster *et al.*, 1991] Scott H. Foster, Elizabeth M. Wenzel, and R. Michael Taylor. Real-time synthesis of complex acoustic environments. In *Proc.* (IEEE) *ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, October 1991. Summary.

[Herder, 1998] Jens Herder. Sound Spatialization Framework. Web site, University of Aizu, Japan, 1998. `http://www-ci.u-aizu.ac.jp/SF/`.

[Ishikawa, 1999] Kimitaka Ishikawa. Using a MIDI module as a sound spatialization backend. Master's thesis, University of Aizu, 1999.

[Kendall and Martens, 1988] Gary Kendall and William L. Martens. Spatial reverberation. U.S. Patent 4,731,848; European patent specification 0 207 084 B1, 1988.

[Meyer *et al.*, 1965] E. Meyer, W. Bugtorf, and P. Damaske. An apparatus for electroacoustical simulation of sound fields: Subjective auditory effects at the transition between coherence and incoherence. *Acustica*, 15:339–344, 1965.

[Rasch and Plomp, 1984] R. A. Rasch and R. Plomp. The listener and the acoustic environment. In D. Deutsch, editor, *The Psychology of Music*, pages 135–147. Academic Press, 1984. ISBN 0-12-213560-1 or 0-12-213562-8.

[Tohyama *et al.*, 1995] Mikio Tohyama, Hideo Suzuki, and Yoichi Ando. *The Nature and Technology of Acoustic Space*. Academic Press, London, 1995. ISBN 0-12-692590-9.

Chapter 7

User Testing

User studies can evaluate and validate the performance of a sound spatialization system, including a sound spatialization backend and a procedure for sound spatialization resource management. Since the developed resource management is based on human perception, an evaluation using objective tests would not measure its effectiveness. Testing system performance using subjective tests is not easy, and difficult to reproduce because the test conditions are difficult to control. Auditory experiments and results are comprehensively described in [Blauert, 1996]. Subjects' abilities to localize sound vary across subjects, experimental conditions and tasks. Localization ability depends on the stimulus surrounding sounds and room features. Performance tests can only be done for specific tasks of certain applications.

For the perceptual space discussed in Chapter 3, which is used for the decision process of the resource assignment, user capabilities must be estimated, confirmed and then generalized. More important than system evaluation is system calibration. This is the process of tuning all system parts to give best performance to the user. Elegant resource management also takes into account the performance capabilities of the spatialization backends.

7.1 Task-based performance tests for sound spatialization backends

User performance is measured for the same applications using different spatialization backends. Besides comparing the performance data for backends, it can be put into relation to user performance in a "natural" environment. Simple tasks can be:

- estimating sound direction

- comparing object distance

Monitoring of user performance is typically done using forced-choice selection or applying a tracking device.

7.2 Evaluation criteria used by subjects

Absolute or physical judgments are not appropriate for experimental responses. The questions must be relative and oriented to subjective dimensions of the sound space [Kendall and Martens, 1984, p. 121]:

Relative direction (azimuth and elevation)

Relative distance (range)

Definition clarity and impression of the size of a sound source

Spaciousness room characteristics, i.e., liveness, size, shape, etc.

Spatial texture changes of the sound perception itself through the environment (i.e., room, other objects like occluders)

7.3 Taxonomy of psychoacoustic validations

7.3.1 Comparing sound spatialization backends to reference recordings

- Binaural recording of a defined stimulus through the ears of the subject or dummy head
- Simultaneous recording (i.e., pick up) of a stimulus as a monaural anechoic reference signal
- Binaural spatialization of a stimulus (using a spatialization backend)
- Alternating presentation to the subject of both binaural versions, using an equalized headphone
- Questioning the subject regarding the difference between the stimuli

7.3.2 Comparing sound spatialization backends to a reference impulse response

For psychoacoustic verification of computer models for binaural room simulation [Pompetzki, 1993], the following procedure was employed:

- Measurement of a reference impulse response using a dummy head
- Convolution of impulse response with a monaural anechoic reference signal
- Binaural spatialization of stimulus (using a spatialization backend)
- Alternating presentation to the subject of both binaural versions, using an equalized headphone (ABA order)
- Questioning the subject regarding the difference between the stimuli

7.3.3 Direct comparison between sound spatialization backends

Objective direct comparison is not an easy task because many parameters are involved which can distort the results of such a test. The output devices of the backends might be different and needs to be harmonized. In case of a loudspeaker system the room influence the result through its reverberation.

- Binaural recording of a defined stimulus through the ears of the subject or dummy head
- Simultaneous recording of the stimulus as monaural anechoic reference signal
- Binaural spatialization of the stimulus using the spatialization backends
- Alternating presentation to the subject of both spatialization backends using equalized headphone to the subject
- Questioning the subject regarding the difference between the stimuli

Figure 7.1 shows a schematic of a test for comparing two different spatialization backends. In practice, such a test can be simplified by recording the produced spatialization.

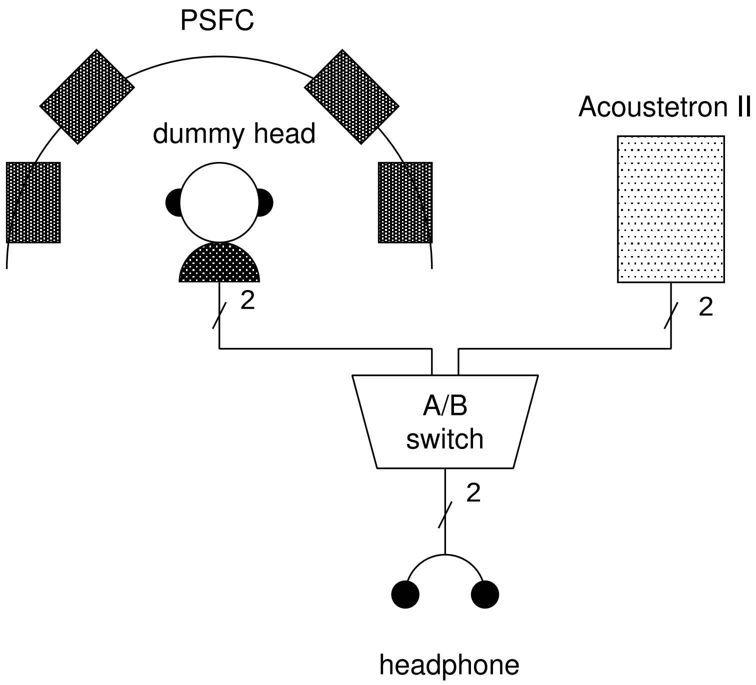


Figure 7.1: A/B test for spatialization backends via dummy head

7.4 Psychoacoustic evaluation of the clustering algorithm

The validity of the presented clustering algorithm in Chapter 4 can be demonstrated for specific configurations. The test falls into the category of the previous section, which directly compares two spatialization backends. Here the same spatialization backend was used, but spatialization was done with and without clustering enabled.

7.4.1 Method

A scene with three sound sources was prepared. A timed script activated the sound sources using MIDI commands. In this evaluation study, the spatialization backend Acoustetron II (see Section 6.1.2) was used. As sound generator, 4 MIDI synthesizer Roland SoundCanvas SC-55mkIIs were used. The sounds were a bird (instrument 124), a telephone (instrument 125), and a gun shot (instrument 128). The sources were triggered with 500 ms delay in between so that the onsets did not overlap. Two of the MIDI synthesizer produced the identical signal for reverberation, which was passed via a mixer to a reverberator Yamaha REV 500 as monaural signal. The reverberant signal mixed with directionalized sound from the spatialization backend. The configuration of the reverberator was setup to simulate a medium sized room. (Parameters were Effect only, 24 ms predelay, 1 s reverb time high-ratio 0.4, and ER level 100.) The reverberator improved externalization [Begault, 1994, p. 97] (see also p. 33 in this thesis) and was used to produce ambient sound for sound sources which could not be spatialized.

stimuli	spatialization channels	number of clusters	ambient sound sources
no restrictions	3	0	0
clustering	2	1	0
ambient	2	0	1

Table 7.1: Stimuli use of spatialization resources

Listening was done using headphones in an anechoic chamber. Five listeners participated. One trial consisted out of a ABA sequence. A stimuli was either three sound source processed with three spatialization channels (N), processed using the developed clustering algorithm (C), or processed using two spatialization channels and one sound source was presented ambient (A).

stimuli	label	x	y	sound
no restrictions	source 1	7.87	-7.87	gun shot
	source 2	7.87	7.87	phone ringing
	source 3	7.87	11.81	bird call
clustering	source 1	7.87	-7.87	gun shot
	cluster 1	7.87	9.84	bird call and phone ringing
ambient	source 1	7.87	-7.87	gun shot
	source 2	7.87	7.87	phone ringing
	ambient	-	-	bird call

Table 7.2: Stimuli source description (using the coordinate system of the CRE API)

This is summarized in Table 7.2 (coordinates are represented using the CRE API [CRE, 1994]). The total number of trials was 72; each stimulus combination was presented 8 times. The nine trial combinations are listed in Table 7.3. The listeners were asked to rate the dissimilarity of the spatial imagery. They marked "1" when the spatial images were judged equal and "5" for the largest difference. The stimuli combinations with itself were included to check if users response is randomly. Also the ratings for a stimuli pair in different order should give similar ratings.

stimulus	abbreviation	first	second
1	NCN	non-restricted	clustered
2	CNC	clustered	non-restricted
3	NAN	non-restricted	ambient
4	ANA	ambient	non-restricted
5	CAC	clustered	ambient
6	ACA	ambient	clustered
7	NNN	non-restricted	non-restricted
8	CCC	clustered	clustered
9	AAA	ambient	ambient

Table 7.3: Trial combinations

7.4.2 Results and discussion

The averaged rating for all lister for each stimulus is shown in Table 7.4. The mean ratings on the diagonal show that not always the listener detected that the same stimulus was presented three times.

		second		
		N	C	A
first	N	1.075	2.225	4.500
	C	2.175	1.000	4.375
	A	4.375	4.525	1.500

Table 7.4: Dissimilarity between intervals: non-restricted (N), clustered (C), and ambient (A)

Average dissimilarity rating regardless the order of listening is shown in Figure 7.2. The average dissimilarity for clustered and ambient processing (CA) was 4.45, for non-restricted to ambient processing (NA) was 4.4375, and for non-restricted to clustered processing was 2.2. The averaged dissimilarly for comparison of all three stimuli with itself was 1.1917.

Sound spatialization with no restrictions in the number of spatialization channels or using clustering for (based on the specific configuration) were rated very dissimilar to the processing with one ambient sound source. The dissimilarity judgments between processing with no restrictions and clustering were ranked half compared to the ratings for ambient processing with the others.

Assuming spatialization resource limitations and specific configuration, processing using clustering improves the spatial imagery. Clustering, as implemented, did not give the same spatial image to processing without limitations.

Bibliography

[Begault, 1994] Durand R. Begault. *3-D Sound for Virtual Reality and Multimedia*. Academic Press, 1994. ISBN 0-12-084735-3.

[Blauert, 1996] Jens Blauert. *Spatial Hearing: The Psychophysics of Human Sound Localization*. MIT Press, revised edition, 1996. ISBN 0-262-02413-6.

[CRE, 1994] Crystal River Engineering, Inc. *CRE_TRON Library Reference Manual*, August 1994. Revision B.

[Kendall and Martens, 1984] Gary S. Kendall and William L. Martens. Simulating the cues of spatial hearing in natural environments. In *ICMC: Proc.*

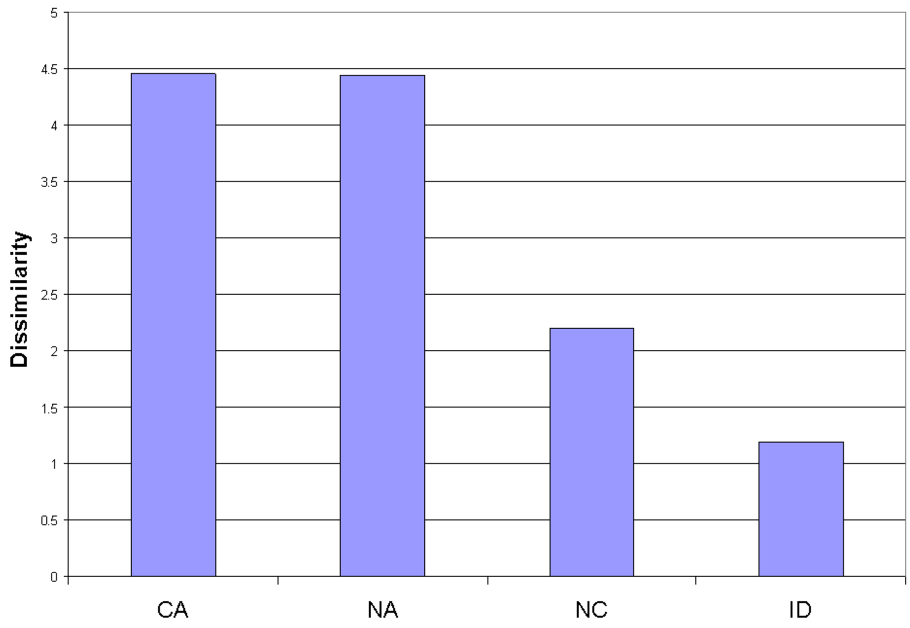


Figure 7.2: Dissimilarity for non-restricted (N), clustered (C), and ambient (A) processing

I*ntnl.* **C***omp.* **M***usic* **C***onf.*, pages 111–126, Paris, 1984. Computer Music Association.

[Pompetzki, 1993] Wulf Pompetzki. *Psychoakustische Verifikation von Computermodellen zur binauralen Raumsimulation.* PhD thesis, Ruhr-Universität Bochum, 1993. (In German).

Chapter 8

Resource Monitoring and Visualization

8.1 Monitoring of resource allocation

Tools for monitoring resource allocation [Herder, 1998] are described in Section 9.2.3. In this section, the allocation for one specific application, the Helical Keyboard [Herder and Cohen, 1996], is shown. A MIDI stream (Fugue from Bach) animates and triggers the animation. Each sound source in the scene corresponds to a note. The interesting parameters are the number of requests by the application for spatialization of sound sources (requested), the number of representative (virtual) sound sources generated through clustering, and the number of active (active) sound sources which are actually sent to the spatialization backend. Source which cannot be processed by the spatialization backend are mixed as ambient sound sources (ambient).

Figure 8.1 shows the resource allocation over a period of around 90 s. Enough spatialization channels are available so that all requested mixels (sound sources) are granted. Virtual (representatives of a cluster) and ambient sources are not allocated. Up to five spatialization channels are requested in this example.

In the next three figures the number of spatialization channels is limited to two. Figure 8.2, shows that the clustering process starts after requests of more than two sound sources. Further, the clustering process can not always prevent the rendering of some sources ambiently. The maximum number of ambient sources is two in cases which the clustering cannot reduce the number of required spatialization channels.

Figure 8.2 shows a short time period (5 s) of Figure 8.2. In this period, the maximum number of ambient sources is one. The number of clusters

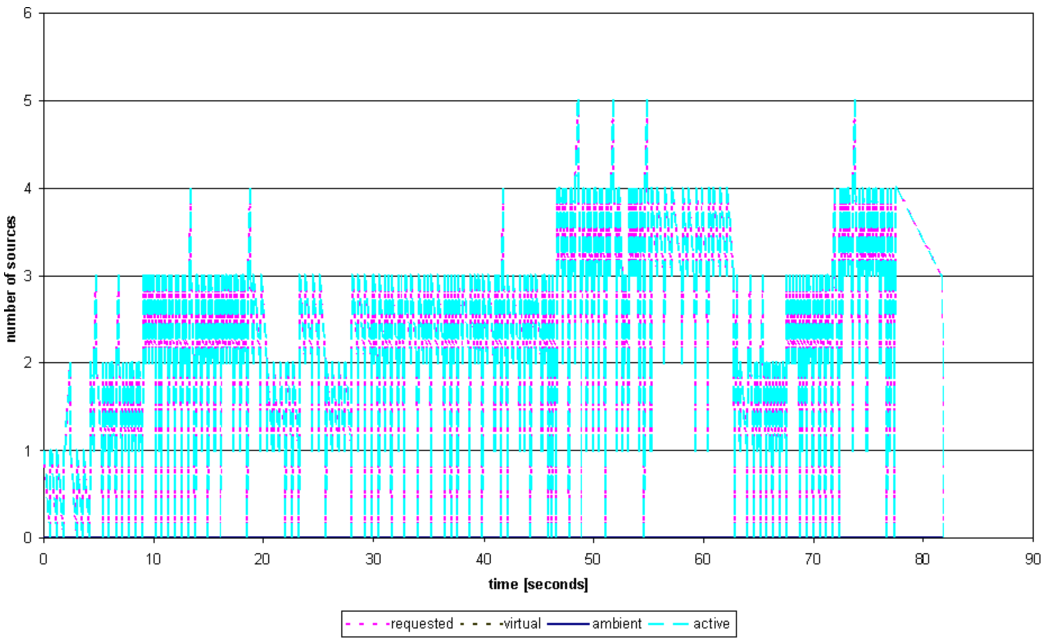


Figure 8.1: Resource allocation without reaching the limited number of spatialization channels

goes up to three, in which one cluster has to processed ambient.

Listener movement changes the parameters for the clustering. This is shown in Figure 8.4, for which listener position was steadily varied on smooth paths around the sound sources. Resource allocation changes more often because of the sound sink movement, but the frame rate is limited so as not to overload the spatialization backend.

8.2 Visualization of the clustering process

In this section the spatialization resource visualizer (described in Section 9.2.5) is used to visualize the clustering process. Again the application is the Helical Keyboard, but this time the sources are activated interactively. The number of spatialization channels is restricted to two. Figure 8.5 shows 88 inactive sound sources and one sound sink.

One sound source becomes active and is highlighted in Figure 8.6. The sound sink is presented with an up vector, a front vector, and a lateral axis, orthogonal to both vectors. This shows the orientation clearly. Additional labels L and R mark the left and right sides from the sink perspective.

Two sound sources are active and highlighted in Figure 8.7. The cluster-

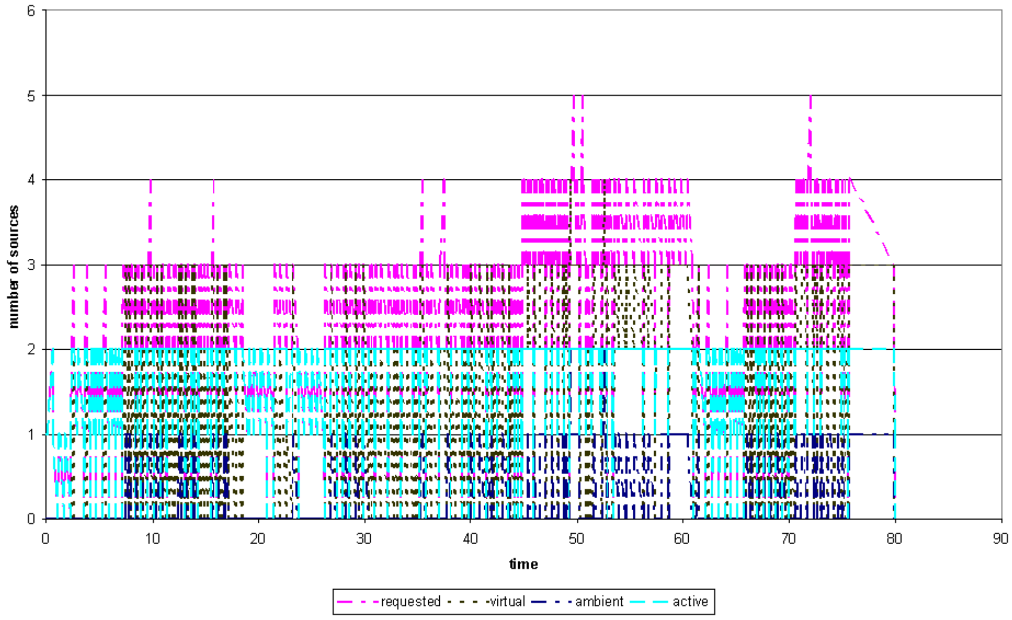


Figure 8.2: Only two spatialization channels are available; clustering process starts, along with source assignment to ambient channels

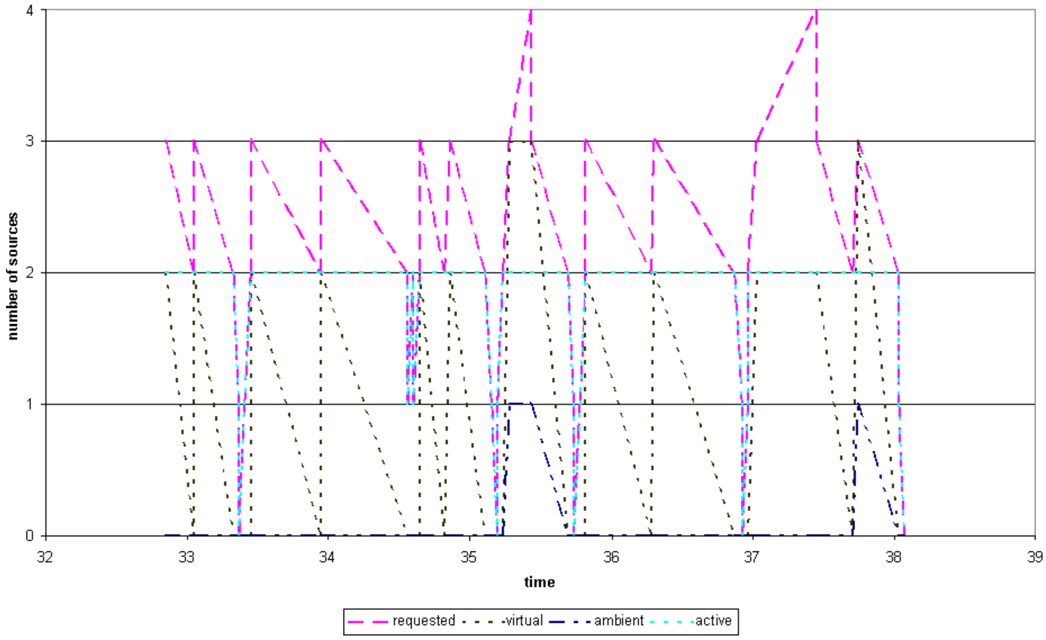


Figure 8.3: Enlargement of a section of Figure 8.2

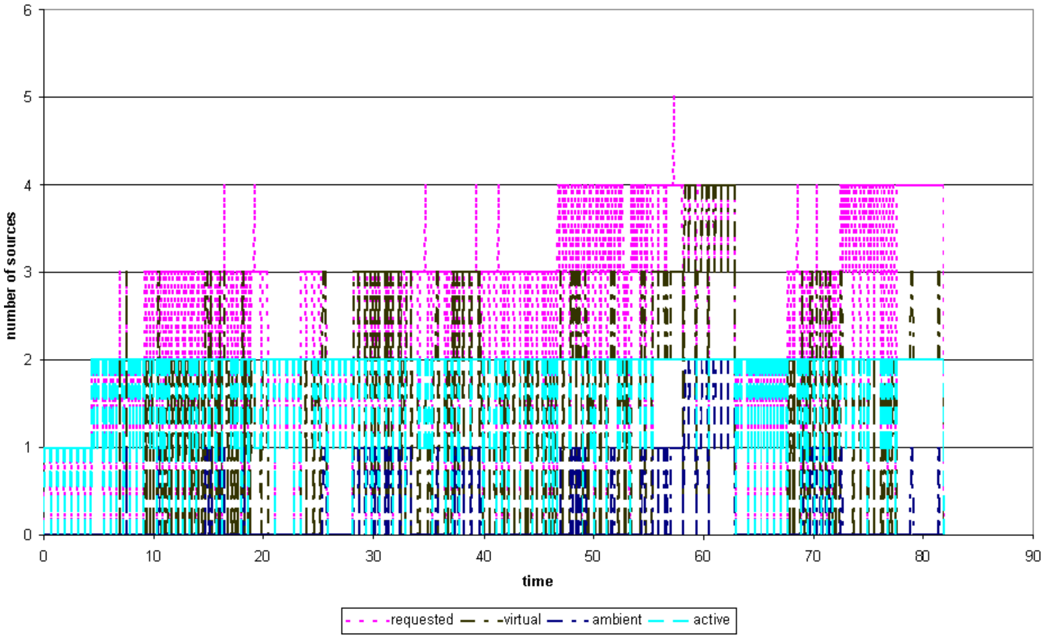


Figure 8.4: Listener movement changes clustering

Figure 8.5: 88 inactive sound sources and one sound sink

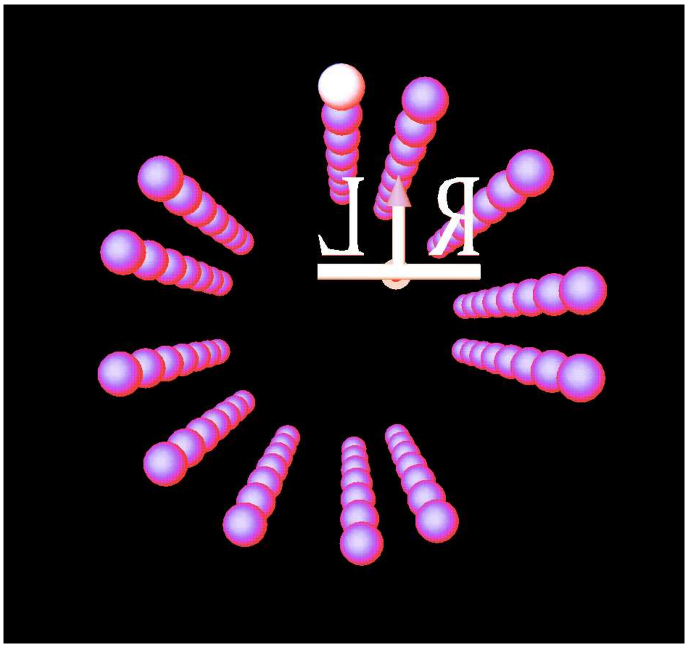


Figure 8.6: One sound source becomes active

ing process starts only when the limit of available spatialization channels is hit. As long enough resources are available, no clustering is required.

In Figure 8.8 three sound sources are requested. The clustering process starts, finding two clusters, one representing two sound sources. The other contains only one sound source. The representative sound sources of each cluster are highlighted. A small tether associates the sound sources in each cluster with their representative sound source.

Four sound sources are requested in Figure 8.9. Two clusters, each with two sound sources, are formed.

Rotation of the sound sink changes the cluster allocation, as shown in Figure 8.10, redistributing the four sources into a single cluster. The resolution cones are listener orientation-dependent. To the side of a listener, localization errors in elevation are lower and localization errors in azimuth higher (see Section 3.8.1).

Moving the sound sink closer to the sources in Figure 8.11 again changes the cluster distribution. Three sound sources on the left are in one cluster. Localization errors in azimuth to the front are small, so that the sound source in the upper front direction cannot be grouped with the other requested sound sources.

Figure 8.12 shows the resolution cones becoming smaller again. Moving

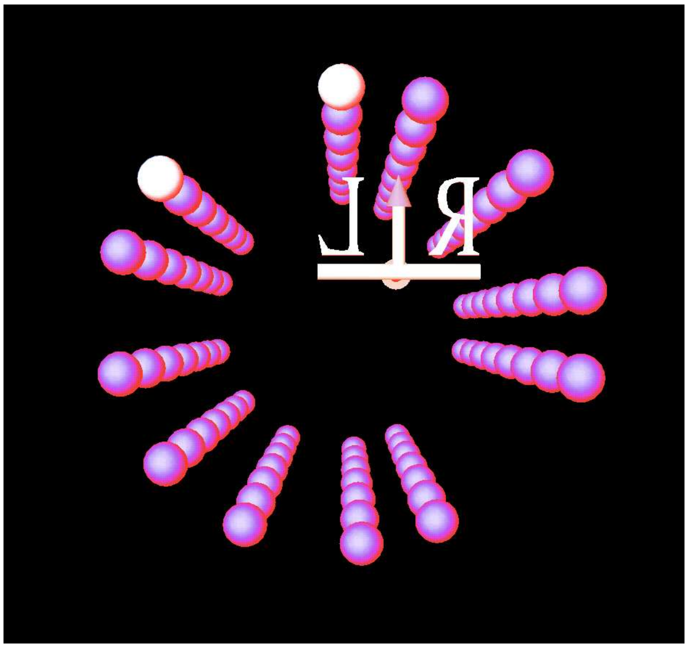


Figure 8.7: Two sound source are requested

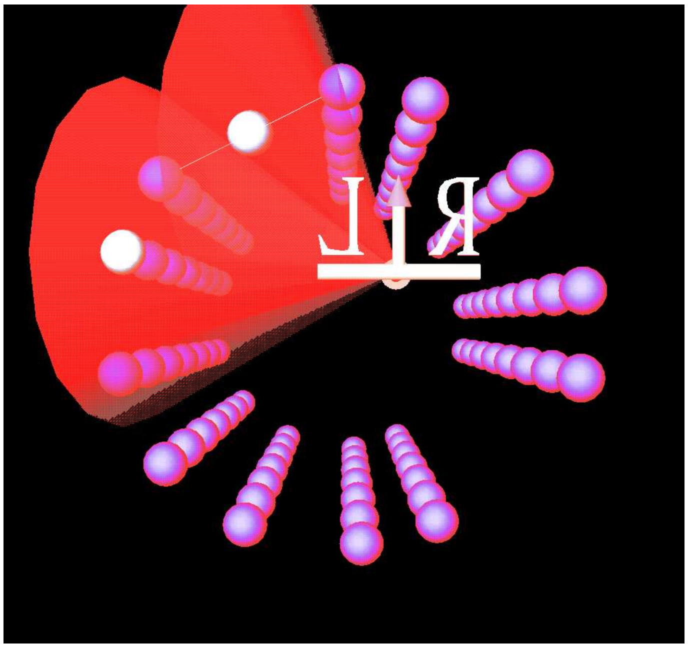


Figure 8.8: Three sound sources are requested; clustering algorithm becomes active

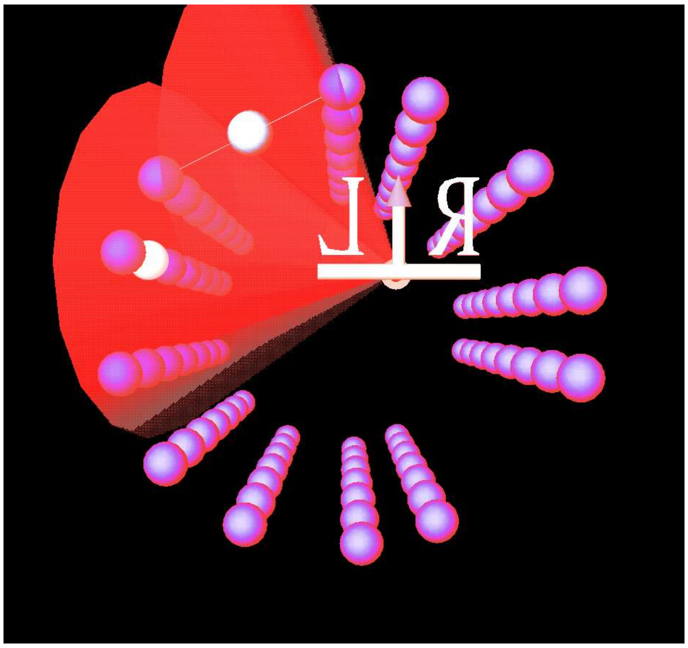


Figure 8.9: Four sound sources are requested

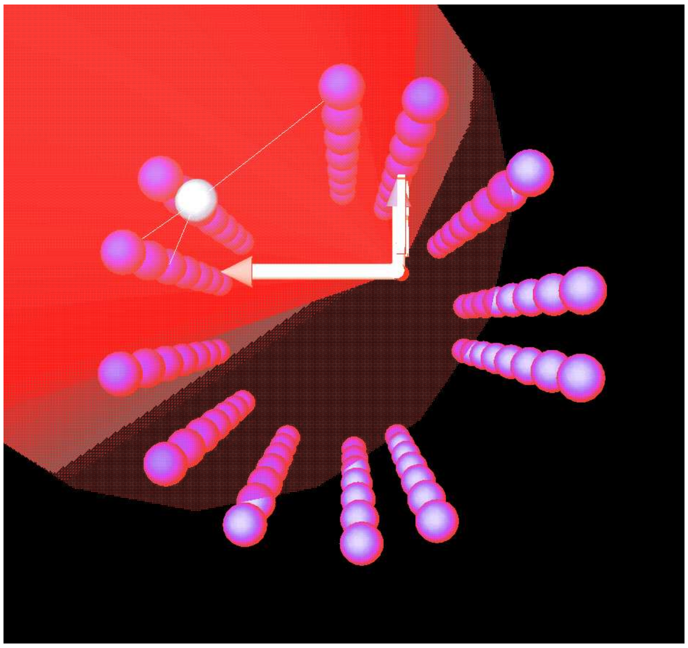


Figure 8.10: Rotation of the sound sink changes the cluster allocation

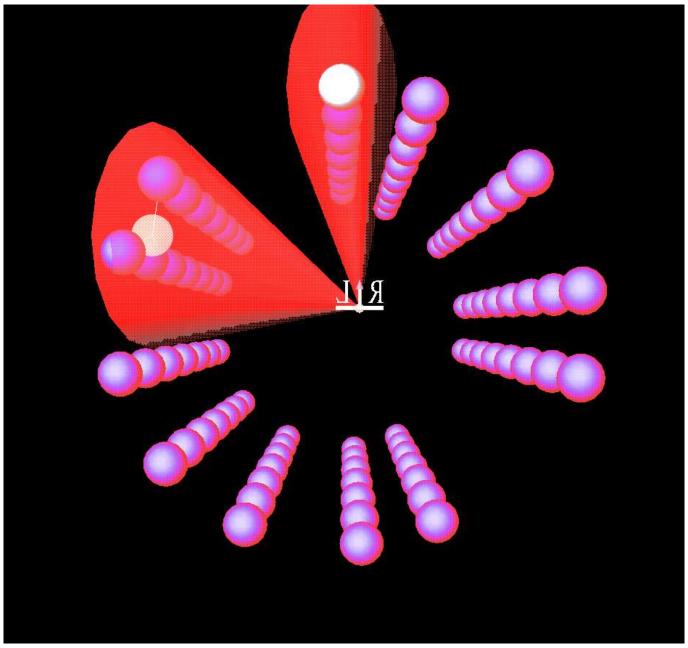


Figure 8.11: Moving closer with the sound sink to all sound sources

the sink closer to the sources changes the number of sound sources which can be clustered. This time three clusters are allocated. One cluster will be processed ambiently.

In Figure 8.13, the clusters get split up after further movement of the sink close to the sound sources. This shows that in the near field of the head, clustering does not reduce the number of required spatialization channels.

In Figure 8.14, three requested sound sources are passed. One active sound source is still in front of the sink. Two sound sources which are now in the back of the sink get clustered. One cluster must be processed ambiently.

The side view in Figure 8.15 shows that the resolution cones to the back of the listener are larger.

Bibliography

[Herder and Cohen, 1996] Jens Herder and Michael Cohen. Design of a Helical Keyboard. In Steven P. Frysinger and Gregory Kramer, editors, ICAD *'96 — Int. Conf. on Auditory Display*, Palo Alto, CA; USA, November 1996.

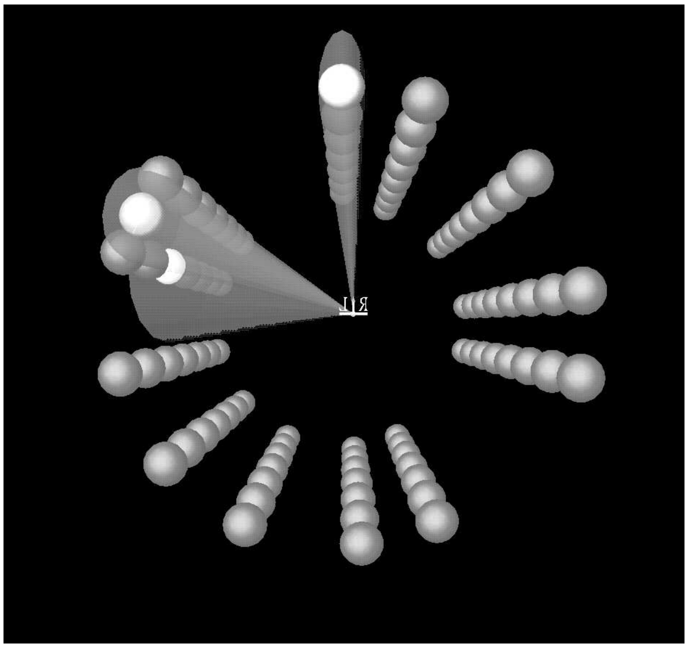


Figure 8.12: Moving closer again; cones become smaller

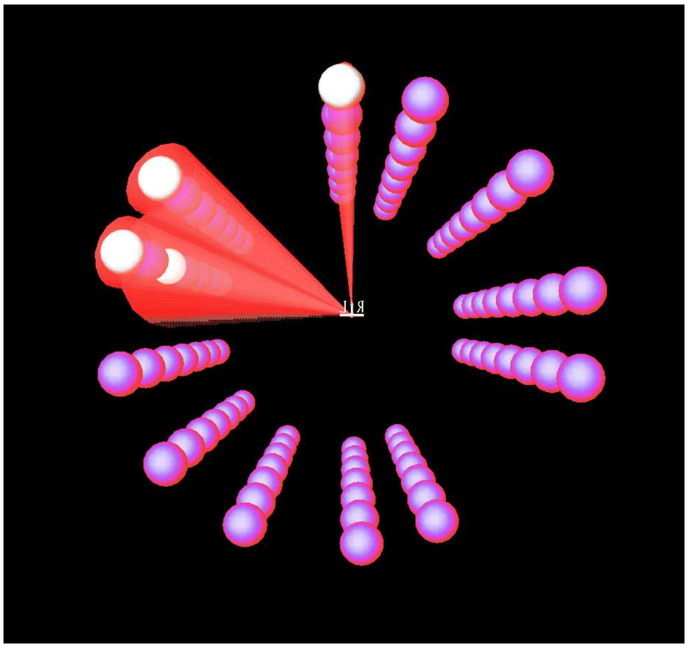


Figure 8.13: Moving closer again; cones become smaller; clusters get split up

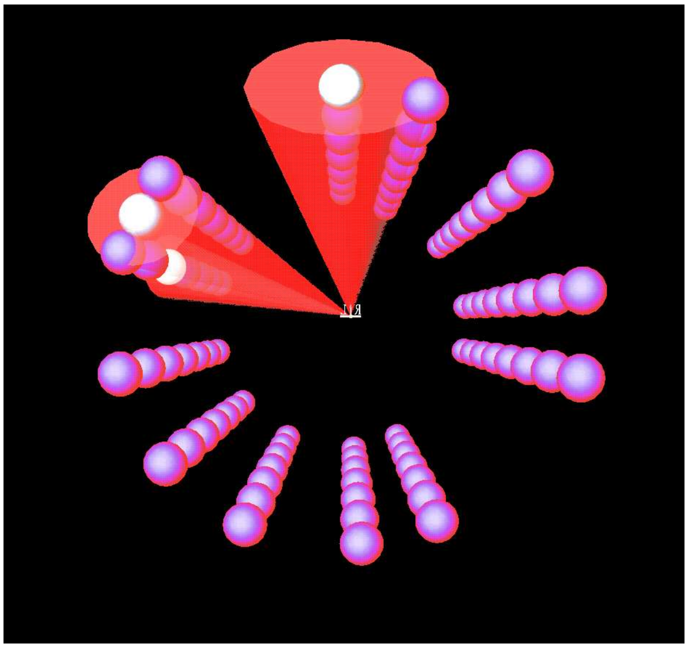


Figure 8.14: Besides on sound source, active sound sources are passed

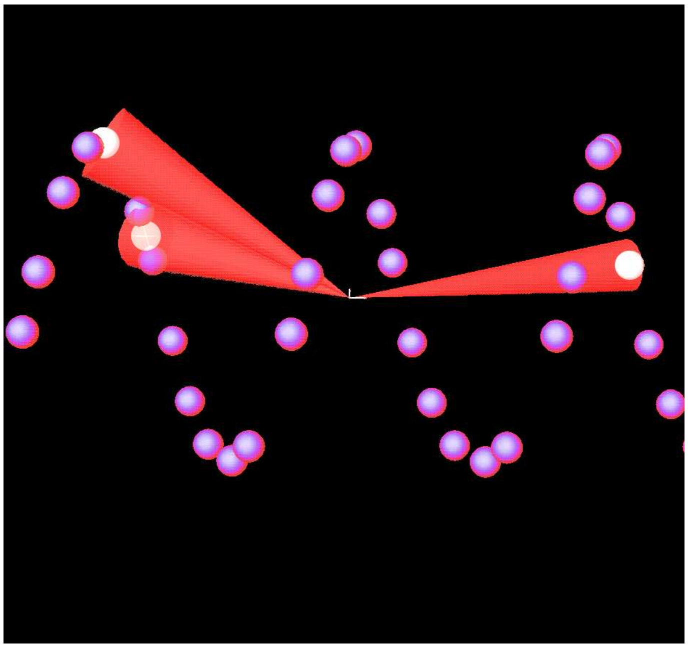


Figure 8.15: Side view shows that resolution cones to the back are larger

[Herder, 1998] Jens Herder. Tools and Widgets for Spatial Sound Authoring. *Computer Networks & ISDN Systems*, 30(20-21):1933–1940, October 1998.

Chapter 9

Sound Spatialization Authoring

Broader use of virtual reality environments and sophisticated animation spawn a need for spatial sound. Until now, spatial sound design has been based very much on experience and trial and error. Most effects are hand-crafted, because good design tools for spatial sound do not exist. This chapter discusses spatial sound authoring and its applications, like shared virtual reality environments based on VRML. New concepts introduced by this research are an inspector for sound sources, an interactive resource manager, and a visual soundscape manipulator. The visual tools are part of a sound spatialization framework and allow a designer/author of multimedia content to monitor and debug sound events. Resource constraints like limited sound spatialization channels can also be simulated.

More and more applications use virtual reality environments with spatial sound as a user interface. The demand for animations with impressive and immersive sound increases, as audio-visual equipment which can produce such effects becomes increasingly available. Spatial sound has migrated from special platforms to everyones' desktop. This is due to better general-purpose processors, which allow spatial sound processing in software, and hardware support, in the form of audio cards.

Wide distribution of content including spatial sound for virtual reality environments over the internet was made possible with the introduction of the Virtual Reality Modeling Language (VRML 2.0 [Bell *et al.*, 1996]). Even though the specification does not cover all aspects of spatial sound, dramatic effects can be produced. Available tools for producing VRML content have some support for spatial sound authoring, but in general they are not sufficient. The research described by this thesis has developed widgets, user interface objects with encapsulated geometry and behavior, to control and display properties of soundscapes and sound objects.

Spatial sound Spatial sound [Anderson and Casey, 1997] is modeled by many attributes. Directionalization allows to point towards a sound source. Distance cues are based on delay, reverberation and loudness. The space defines the reverberation and first (and higher) order reflections.

Authoring: Creativity and engineering For spatial sound authoring, two disciplines converse. An artist or content producer with strong emphasis on creativity brings the ideas or better defines what should be done. On the other hand, skills from engineering are needed to define the space and actually produce the effects.

Requirements Requirements for a spatial sound authoring toolset on which this research has focused are

- soundscape visualization,
- soundscape manipulation,
- sound object visualization,
- sound object editing, and
- sound resource monitoring.

When developing an authoring tool for spatial sound, the underlying system, like the spatial sound API, defines a lot of the functionality and might restrict the generality and portability of the application. It is important to keep in mind that the main task for the author is to develop content, and that rendering issues for different platforms should not interfere. An example of modeling rendering issues was given in [Brown and Allard, 1997]: the attenuation of the frequency spectrum of a waterfall (part of the "Jungle Island" demonstration) was modeled by using two sound sources with different audible range and frequency band. This made it possible to hear the rumbling of the waterfall in the distance as well as the high frequency components when sufficiently close. The effect was impressive, but what if the sound renderer supports distance frequency attenuation? Would it be better if the API and also the sound authoring tools hid those steps from the user? The same approach can be taken for sound occluders and first-order reflections.

9.1 Previous research

9.1.1 Spatial sound application programmer interfaces

The latest VRML [Bell *et al.*, 1997] specification has only a sound node to support spatial sound, and does not define soundscape attributes to describe reverberation. A browser might guess the size of the space and then set important reverberation parameters for sound spatialization. The Java3D [Sowizral *et al.*, 1997] specification is in that regard more advanced, supporting a notion of a soundscape, an application area with aural attributes capturing delay times and reflections. The discussed APIs are good for multimedia content, but are not suitable for room acoustics, which are much more complicated and require a more physical approach — specification of the material and transfer functions of any sound object (e.g., a wall) in the simulated space. On the other hand such simulations are not yet done in realtime. Realtime processing becomes possible if the room parameters are processed beforehand.

9.1.2 Spatial sound authoring systems

Most spatial sound authoring systems are closed and do not allow users to develop content for different backend configurations. A multiple audio window system [Cohen, 1993] gives each user a visual, exocentric view on a scene and allows realtime interaction and sound object editing based on direct manipulations and cut & paste metaphors.

In shared virtual environments (e.g., AlphaWorld) [Waters and Barrus, 1997], users not only explore and meet, they also extend and build the space which they inhabit. Building such a space which is part of a larger system includes sound. The restrictions/constraints in such a case are even tighter, to avoid the social infrastructure becoming damaged through the creation of areas which are inaccessible due to resource load on either the server or client side.

9.2 Sound spatialization development environment

The developed environment consists of a library to manage all sound processing, a visual soundscape controller (to handle mapping between geometric application space and soundscape), a sound resource allocation monitor, a

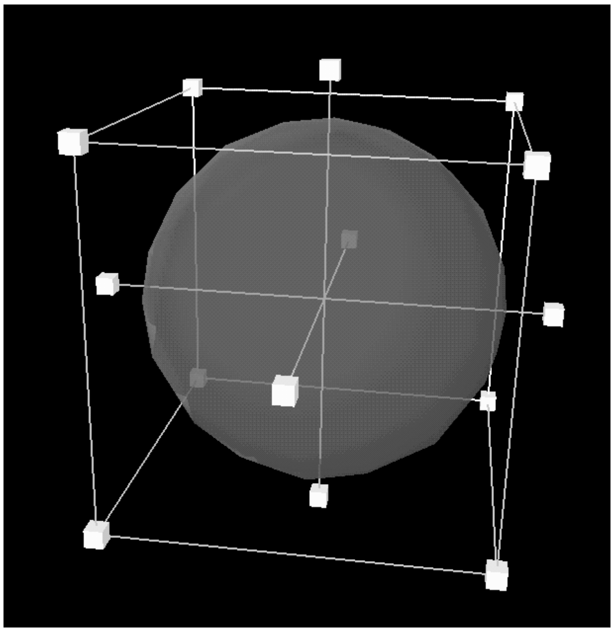


Figure 9.1: Soundscape deformer

soundscape visualizer, and an editor for sound objects. The following subsections introduce the modules and the data-flow.

9.2.1 Soundscape control

During the design of the helical keyboard [Herder and Cohen, 1996], we became aware that global control of the mapping between visual space and acoustical space could improve the intended experience. A major part in the design was the soundscape. All keys had to be differentiable by location, mainly direction. The helix itself became visually a very tall object. If the listener is placed in the center, then keys playing in the far upper part or lower part could not be well differentiated by azimuth, and also the volume for them was too low. As a solution to these problems we developed the soundscape deformer, a 3D widget that controls the scene space $\rightarrow$ soundscape mapping. The scene space can be shifted around, which induces a translation of the soundscape. In that regard the soundscape deformer can be seen as a generalization of stereo panning for 3D (e.g., balance potentiometer of an amplifier, also known as a pan pot).

The soundscape deformer, shown in Figure 9.1, provides a visual representation of a linear mapping. A sphere in the center represents the case in

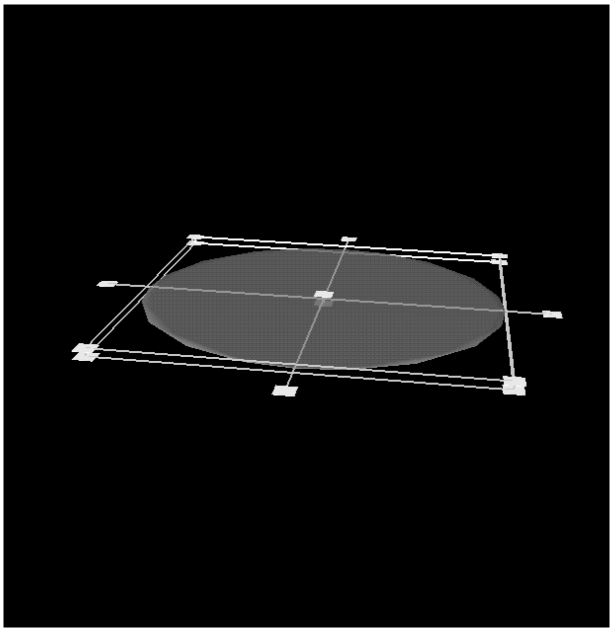


Figure 9.2: Soundscape deformer: flattening

which the scene space is directly mapped to the soundscape. Figure 9.2 shows the soundscape reduced in height, flattening the spatial audio position of all sound objects to a plane. Another example, shown in Figure 9.3, reduces the horizontal dimension, compressing the left↔right attribute. As an extreme example, shown in Figure 9.4, the sphere can be reduced to a point, giving a diotic soundscape, in which all objects seem to be at one place inside the user's head.

9.2.2 Portable content: Authoring for different platforms

A commercial application or multimedia product might be required to run on different platforms. Those platforms should be taken into consideration when doing spatial sound authoring.

Output devices Backends vary; a system can use loudspeakers in single, stereo, stereo with crosstalk cancelation, and array [Amano *et al.*, 1998] configurations. Headphones or nearphones are also widely used. Across these devices the amount of immersion or believable illusion differs drastically. Despite the fact that most people don't have an absolute tone hearing, the

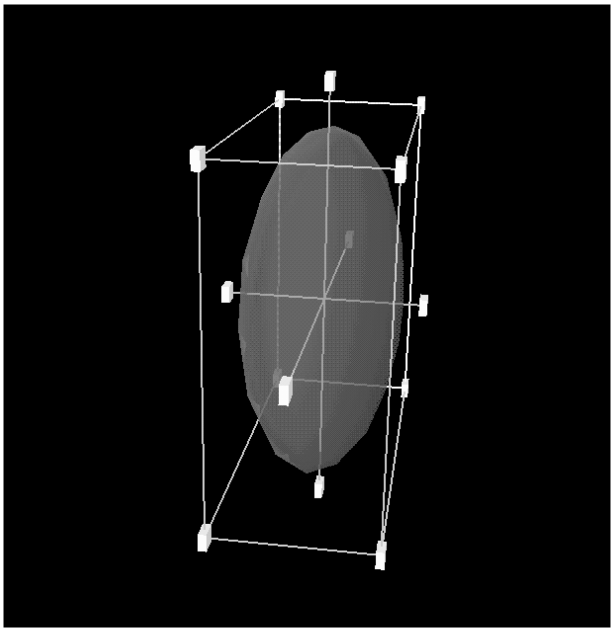


Figure 9.3: Soundscape deformer: narrowing

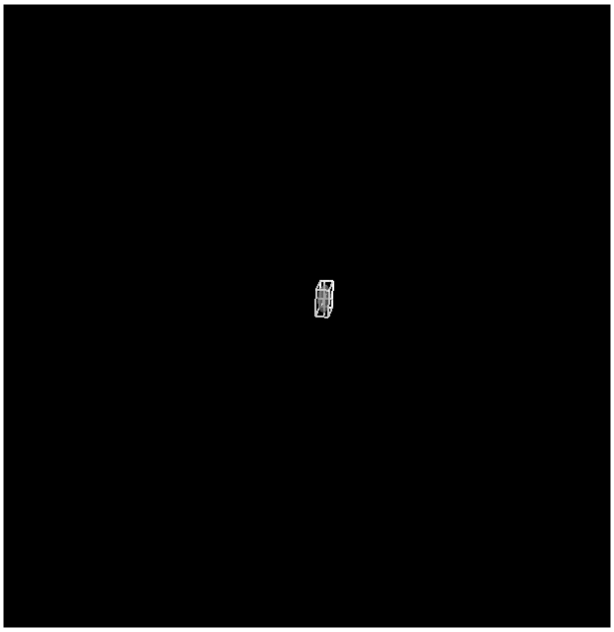


Figure 9.4: Soundscape deformer: extreme diotic case

frequency spectra of the output device needs to be adjusted and considered[1]. If a headphone cannot produce a rumbling sensation in the stomach, then the author who creates multimedia content for a broad range of platforms needs to remember that, and might choose a different or additional acoustical event.

Spatialization backends The design of spatialization backends depends on the output devices surveyed in the paragraph above. Part of the spatialization process involves processing filter functions (convolution) and reverberation. Such processing can be done either in hardware [Wenzel *et al.*, 1990] and software [Intel, Inc., 1997]. In the later case the main CPU load might increase unacceptably if the spatial sound design did not anticipate such problems. Otherwise the software for spatialization disables resource allocation and the acoustical effect cannot be achieved.

Sound processing Sound processing — in the form of audio (e.g., wav) files, MIDI synthesis, or physical models — may use system resources and compete with other processes like the spatialization. A good system maintains balance and optimizes for the user based on psychoacoustic metrics.

9.2.3 Monitoring sound resource allocation

How can the above mentioned problems be addressed during the process of spatial sound authoring? One solution, but impractical, is to have all platforms available and to do tests, but even so not all configurations can be covered.

We propose to monitor during the authoring process the resource requests. This will help to inform the author and sound developer about active sources and resource allocation. We have developed a sound spatialization resource manager [Herder and Cohen, 1997] including a monitor for the requests and allocations. The panel shown in Figure 9.5 gives access to the number of sound sources and sinks, the number of active (i.e., requested) sources, the number of ambient sources, and the number of virtual sources in a scene. Virtual sources represent a cluster of sound sources which can be spatialized as a single source. The audio signals are mixed before the spatialization takes place. Ambient sources do not use spatialization resources, but produce a load for the sound generation.

[1]For example, the head-related transfer function needs to be equalized for the headphone in use. Even better would be individual HRTFs.

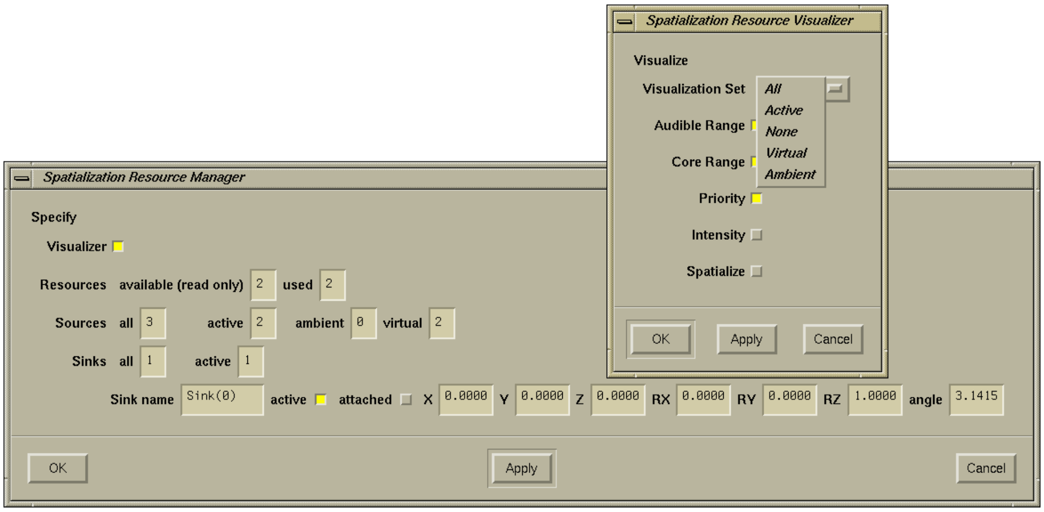


Figure 9.5: Sound spatialization resource manager panel

9.2.4 Simulating resource allocation via resource constraints

How would a certain kind of content be produced on a system with few spatialization channels? The spatialization resource manager resources (i.e., the number of spatialization channels) can be dynamically constrained by interacting with the control panel (as seen in Figure 9.5). This allows monitoring of resource allocation across finite capabilities. In the same way of course it is made audible and allows the designer to compare different configurations.

9.2.5 Spatialization resource visualizer

The spatialization resource visualizer is an inspector for sound objects in the soundscape, a visual debugger for sound objects in virtual reality environments. These are sound sources and sinks, generalization of listener and microphone. Figure 9.6 shows on the left side a test scene with the corresponding sound objects in the visualizer on the right side. A special case involves virtual sound sources which do not exist in the virtual reality environment scene and are generated during the clustering process of the spatialization resource manager. The set of sound sources can be selected (using the preference menu, seen in Figure 9.5) to focus on all, active, virtual, or ambient sound sources. A sound source can be displayed using its core range, which is an ellipsoid representing a zone with maximum intensity [Bell *et al.*, 1997], and its audible range, shown as a translucent ellipsoid representing the space in which the source is audible. Between the two ranges the intensity drops off according to the square of the distance. Priority and intensity of the sound node may be included as text values facing the user. Different states of a sound node can be conveyed using color codes for the core range.

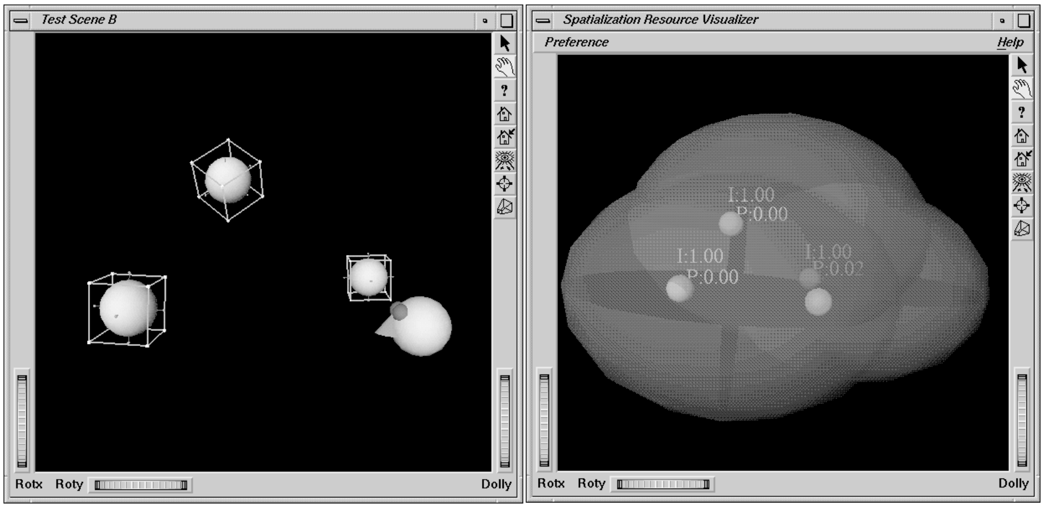


Figure 9.6: Test scene with sound source visualization

9.2.6 Sound source editor

The editor shown in Figure 9.7 allows one to edit and monitor sound nodes [Bell *et al.*, 1997] in a virtual reality runtime environment. The user invokes the editor via mouse click on a sound node in the spatialization resource visualizer.

A source radiation pattern can be defined by a core range and audible range, represented by the sound node fields `minBack`, `minFront`, `maxBack`, and `maxFront`. The resource allocation algorithm uses the `priority` value to rank the sources. The fields `direction` and `location` change orientation and position of the sound node in its local coordinate space. Changes are immediately manifested in the spatialization resource visualizer. If the field values are modified during runtime by the application, the fields in the attached editor are updated. This allows textual sound behavior monitoring of a scene.

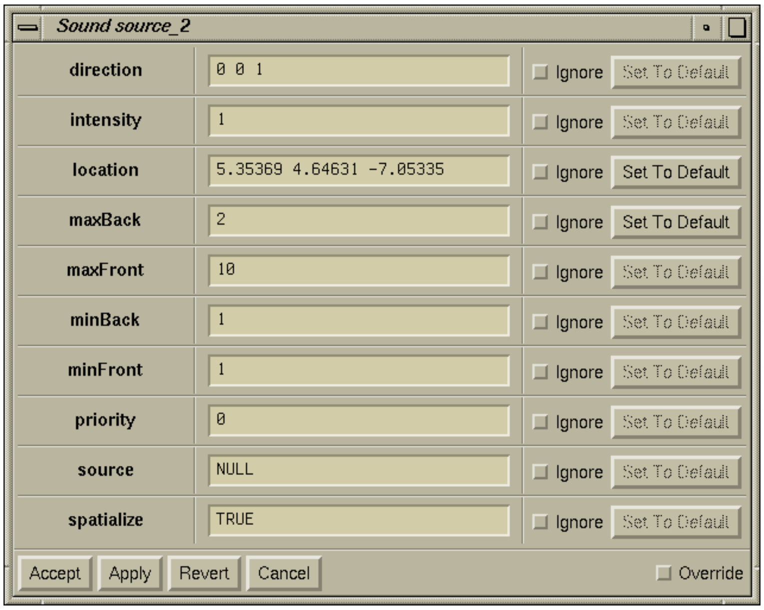


Figure 9.7: Sound node editor

9.2.7 Tool data-flow

Figure 9.8 shows the data-flow between the system components. All tools keep each other up-to-date. Changes from the virtual reality environment propagate to the sound node editor directly. Requests for sound resources are processed by the sound spatialization resource manager and then visualized by the Spatialization Resource Visualizer (seen on the right side of Figure 9.6). The user can select a resource in the visualizer and invoke the sound node editor for the associated sound node. A change here would propagate back to the visualizer via the runtime environment and resource manager. The resource manager updates the panel, so that numeric information about the resource allocation process is available. The panel can also change parameter of the allocation process, which will also propagate through the tools.

9.2.8 Implementation

Our prototype was developed on an SGI Indigo 2 Extreme, connected to an Acoustetron II from Aureal/Crystal River Engineering and Roland Sound Modules. The Open Inventor graphics toolkit [Wernecke, 1994] was expanded for classes (nodes) to support the spatial sound extensions, which were used

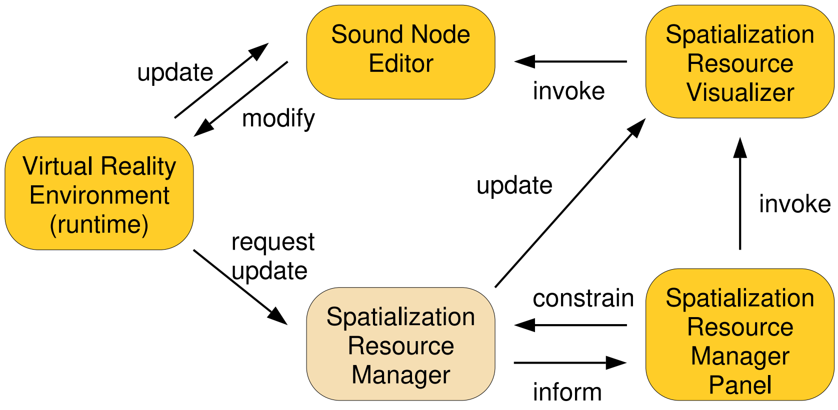


Figure 9.8: Tool data-flow

for our virtual reality applications. Open Inventor is a superset of the VRML 1.0 standard [Bell *et al.*, 1995], which does not support sound or dynamic behavior of objects. For the sound extensions of Open Inventor, we followed the VRML97 standard [Bell *et al.*, 1997], but added a node for sound sinks. This allows to have multiple sinks and a sink, which can be separated from the viewpoint.

Bibliography

[Amano *et al.*, 1998] Katsumi Amano, Fumio Matsushita, Hirofumi Yanagawa, Michael Cohen, Jens Herder, William Martens, Yoshiharu Koba, and Mikio Tohyama. A Virtual Reality Sound System Using Room-Related Transfer Functions Delivered Through a Multispeaker Array: the PSFC at the University of Aizu Multimedia Center. *TVRSJ:* **Trans**. *of the* **V***irtual* **R***eality* **S***ociety of* **J***apan*, 3(1):1–12, March 1998. ISSN 1342-4386.

[Anderson and Casey, 1997] David B. Anderson and Michael A. Casey. The sound dimension. *IEEE Spectrum*, 34(3):46–50, March 1997.

[Bell *et al.*, 1995] Gavin Bell, Anthony Parisi, and Mark Pesce. The Virtual Reality Modeling Language, Version 1.0 Specification, May 1995. `http://www.vrml.org/Specifications/VRML1.0/`.

[Bell *et al.*, 1996] Gavin Bell, Rikk Carey, and Chris Marrin. The Virtual Reality Modeling Language, Version 2.0 Specification, ISO/IEC CD 14772, August 1996. `http://www.vrml.org/VRML2.0.old/`.

[Bell *et al.*, 1997] Gavin Bell, Rikk Carey, and Chris Marrin. ISO/IEC 14772-1:1997: The Virtual Reality Modeling Language (VRML97), 1997. `http://www.vrml.org/Specifications/VRML97/`.

[Brown and Allard, 1997] Geoff Brown and Ed Allard. Sound Bytes: VRML Authoring For Noisy Worlds. In *SIGGRAPH Course Notes*. The Association for Computing Machinery, August 1997.

[Cohen and Koizumi, 1998] Michael Cohen and Nobuo Koizumi. Virtual gain for audio windows. *Presence: Teleoperators and Virtual Environments*, 7(1):53–66, February 1998. ISSN 1054-7460.

[Cohen, 1993] Michael Cohen. Throwing, pitching, and catching sound: Audio windowing models and modes. *IJMMS: the Journal of Person-Computer Interaction*, 39(2):269–304, August 1993. ISSN 0020-7373.

[Herder and Cohen, 1996] Jens Herder and Michael Cohen. Design of a Helical Keyboard. In Steven P. Frysinger and Gregory Kramer, editors, ICAD*'96 — Int. Conf. on Auditory Display*, Palo Alto, CA; USA, November 1996.

[Herder and Cohen, 1997] Jens Herder and Michael Cohen. Sound Spatialization Resource Management in Virtual Reality Environments. In *ASVA'97 — Int. Symp. on Simulation, Visualization and Auralization for Acoustic Research and Education*, pages 407–414, Tokyo, Japan, April 1997. The Acoustical Society of Japan (ASJ).

[Intel, Inc., 1997] Intel, Inc. Intel Realistic Sound Experience (3D RSX). White paper, 1997. `http://developer.intel.com/ial/rsx/WPAPER.HTM`.

[Sowizral *et al.*, 1997] Henry Sowizral, Kevin Rushforth, Michael Deering, Warren Dale, and Daniel Petersen. Java^{TM} 3D API Specification. Sun Microsystems, August 1997. `http://www.javasoft.com/products/java-media/3D/forDevelopers/3Dguide/j3dTOC.doc.html`.

[Waters and Barrus, 1997] Richard C. Waters and John W. Barrus. The rise of shared virtual environments. *IEEE Spectrum*, 34(3):20–25, March 1997.

[Wenzel *et al.*, 1990] Elizabeth M. Wenzel, Philip K. Stone, Scott S. Fisher, and Scott H. Foster. A system for three-dimensional acoustic "visualization" in a virtual environment workstation. In *Proc. First* IEEE *Conf. on Visualization*, pages 329–337, San Francisco, October 1990.

[Wernecke, 1994] Josie Wernecke. *The Inventor Mentor*. Addison-Wesley, 1994. ISBN 0-201-62495-8.

Chapter 10

Conclusion

10.1 Summary

The Sound Spatialization Resource Management Framework was tested with three different kinds of sound spatialization backends, based on different technologies including HRTFs, stereo intensity panning, and loudspeaker array sound field control. The Sound Spatialization Resource Management Framework was deployed and tested in the Multimedia Center at the University of Aizu, which features the PSFC system, with more than 15 loudspeakers.

An overview of different applications (e.g., chatspace, Helical Keyboard) which require sound spatialization were presented. Based on those, requirements were developed and reflected in the algorithms presented in this thesis.

A simple resource management algorithm was developed and then extended for clustering of sound sources. The clustering algorithm applies psychoacoustic data (localization errors dependent on listener orientation) as optimization criteria. The advantages and disadvantages were discussed. The developed sound spatialization resource manager improves spatialization fidelity under runtime constraints. Application programmers and virtual reality scene designers are freed from the burden of assigning and predicting the sound sources. The system has been verified with a virtual world animated by a polyphonic MIDI stream.

The extension of resource management to include obstructions like occluders and reflectors in realtime environments was also presented. Furthermore, a discussion describes how to compare, test, optimize, and calibrate sound spatialization backends and controlling algorithms. Finally, a section explains how to do sound spatialization authoring, including steering and predicting resource management, based on tools that are part of the system developed by the author. Novel authoring tools are a soundscape deformer, a 3D widget that controls the scene space $\rightarrow$ soundscape mapping, and an inspector for sound objects in the soundscape, which is a visual debugger for sound objects in virtual reality environments. The tools enable spatial sound authoring for multimedia content. The system can be easily integrated into a virtual reality authoring system and its principles are widely applicable.

10.2 Future research

10.2.1 Visualization

A module for sound source tracing using trajectories and visualizing the convex hull to find out if a source covers a given area for the dynamic behavior could further enhance the system regarding authoring and studying requirements of VR applications. Also a module to visualize statistical data like average orientation, maximum velocity, average velocity, maximum intensity, average intensity, average audible range, etc., would complete the system. Such visual tools could further enhance the spatial sound authoring process.

10.2.2 Visual editing

The editing features of sound objects in the authoring part of the system are restricted to textual input. A better user interface would incorporate manipulators for the different object attributes to allow instantaneous object changes.

10.2.3 Abstract spatialization backend interface — fine grained control for frequency range requirements

The introduced interface for spatialization backends in Chapter 6 provides only a coarse control. A fine grain control would allow better resource management with less computational costs. The specified interface is strongly influenced by the interfaces of available spatialization backends.

A better system would allow to pass for each spatialization channel the frequency range and application specific resolution requirements. If the frequency range is low then the audio renderer might not calculate for different elevation because those cannot be distinguished by a user in such cases. Also the sampling rate could be reduced.

10.2.4 Control over distance modeling to enhance clustering

The clustering algorithm described in Chapter 4 can be further enhanced if the resource manager gets control over the distance modeling before mixing of sound sources for a representative virtual sound source is done. Then all sound sources on one resolution cone can be clustered together regardless of their distance from a sound sink. In the described algorithm, sound sources are not clustered together if the distance is larger than a certain perceptual threshold. Good distance modeling requires reverberation handling separately from distance attenuation (see Section 3.4). Usually, distance modeling is done by the spatialization backend and is a rendering issue.

10.2.5 Control over Doppler shift rendering to enhance clustering

In the clustering algorithm of Chapter 4, sources are not clustered together if their to be generated Doppler shift (i.e., velocity and moving direction) differs more than a specified threshold. This is necessary because the Doppler shift rendering is done in the spatialization backend. A different approach could apply the Doppler shift rendering before mixing the sound sources. In such case, no Doppler shift would be calculated for the virtual (i.e., representative) sound source. More sound sources could be clustered and the overall system performance enhanced, similar to the suggestion in the previous Section 10.2.4.

10.2.6 Standard tests for spatialization backends

For testing and comparing sound spatialization systems, standard tests are required, as are already done for graphics systems. Those tests will enable better calibration of the backends.

Index

Appendix A

Author's Publications

List of the author's publications in the field of computer science.

Bibliography

[1] Katsumi Amano, Fumio Matsushita, Hirofumi Yanagawa, Michael Cohen, Jens Herder, Yoshiharu Koba, and Mikio Tohyama. PSFC: the Pioneer Sound Field Control System at the University of Aizu Multimedia Center. In RO-MAN *'96 - 5th* IEEE *International Workshop on Robot and Human Communication*. IEEE, November 1996.

[2] Katsumi Amano, Fumio Matsushita, Hirofumi Yanagawa, Michael Cohen, Jens Herder, William Martens, Yoshiharu Koba, and Mikio Tohyama. A Virtual Reality Sound System Using Room-Related Transfer Functions Delivered Through a Multispeaker Array: the PSFC at the University of Aizu Multimedia Center. *TVRSJ:* **Trans**. *of the* **V***irtual* **R***eality* **S***ociety of* **J***apan*, 3(1):1–12, March 1998. ISSN 1342-4386.

[3] Kiel Christianson and Jens Herder. Mini-lectures in Computer Science on the WWW. University of Aizu, Center for Language Research 1995 Annual Review, December 1995. `http://www-ci.u-aizu.ac.jp/~herder/publications/clr-report95csm/clrreport95csm.html`.

[4] Michael Cohen and Jens Herder. Symbolic representations of exclude and include for audio sources and sinks: Figurative suggestions of mute/solo & cue and deafen/confide & harken. In *Virtual Environments 98*, pages 95/1–4, Stuttgart, June 1998.

[5] Michael Cohen and Jens Herder. Symbolic representations of exclude and include for audio sources and sinks: Figurative suggestions of mute/solo & cue and deafen/confide & harken. In M. Göbel, J. Landauer, U. Lang, and M. Wapler, editors, *Virtual Environments '98, Proceedings of the Eurographics Workshop*, pages 235–242, Stuttgart, Germany, June 1998. Springer-Verlag/Wien. ISBN 3-211-83233-5.

[6] Jens Herder. HiLDTe - Konzepte und Implementierung einer Textur-Synthese Sprache. Studienarbeit, abstract in Computer Graphik Topics 3/90, 1990.

[7] Jens Herder. Konzeption, Implementierung und Integration einer Komponente zur inkrementellen Bezeichner- und Operatoranalyse innerhalb des PSGs. Diplomarbeit (master thesis in German), Technische Hochschule Darmstadt, 1991.

[8] Jens Herder. Cooperative tools for teaching: an impact of a network environment. Annual Report 1996 of the Information Systems and Technology Center, University of Aizu, 1997.

[9] Jens Herder. Tools and Widgets for Spatial Sound Authoring. In Harold P. Santo, editor, *Compugraphics '97, Sixth Int. Conf. on Computational Graphics and Visualization Techniques: Graphics in the Internet Age*, pages 87–95, Vilamoura, Portugal, December 1997. GRASP. ISBN 972-8342-02-0.

[10] Jens Herder. Sound spatialization framework: An audio toolkit for virtual environments. In *First Int. Conf. on Human and Computer*, Aizu-Wakamatsu, Japan, September 1998. University of Aizu.

[11] Jens Herder. Sound spatialization framework: An audio toolkit for virtual environments. *Journal of the 3D-Forum Society, Japan*, 12(9):17–22, September 1998.

[12] Jens Herder. Tools and Widgets for Spatial Sound Authoring. *Computer Networks & ISDN Systems*, 30(20-21):1933–1940, October 1998.

[13] Jens Herder and Michael Cohen. Design of a Helical Keyboard. In Steven P. Frysinger and Gregory Kramer, editors, ICAD *'96 — Int. Conf. on Auditory Display*, Palo Alto, CA; USA, November 1996.

[14] Jens Herder and Michael Cohen. Enhancing perspicuity of objects in virtual reality environments. In *CT'97 — Second Int. Cognitive Technology*

Conf., pages 228–237. IEEE, IEEE Press, August 1997. ISBN 0-8186-8084-9.

[15] Jens Herder and Michael Cohen. Sound Spatialization Resource Management in Virtual Reality Environments. In *ASVA'97 — Int. Symp. on Simulation, Visualization and Auralization for Acoustic Research and Education*, pages 407–414, Tokyo, Japan, April 1997. The Acoustical Society of Japan (ASJ).

[16] Jens Herder and Michael Cohen. The Helical Keyboard — Another Perspective for Virtual Reality and Music. *IJVR - International Journal for Virtual Reality*, 1999. In press.

[17] Jens Herder, Karol Myszkowski, Tosiyasu L. Kunii, and Masumi Ibusuki. A Virtual Reality Interface to an Intelligent Dental Care System. In Suzanne J. Weghorst, Hans B. Sieburg, and Karen S. Morgan, editors, *Medicine Meets Virtual Reality 4*, volume 29 of *Studies in Health Technology and Informatics*, Van Diememstraat 94, 1013 CN Amsterdam, Netherlands, January 1996. IOS Press.

[18] Jan Hesse, Rainer König, Filippo Logi, and Jens Herder. A Prototype of an Interface Builder for the Common Lisp Interface Manager — CLIB. *ACM Sigplan Notices*, 28(8):19–28, August 1993.

[19] Kimitaka Ishikawa, Minefumi Hirose, and Jens Herder. A sound spatialization server for a speaker array as an integrated part of a virtual environment. In IEEE *YUFORIC Germany 1998*, Stuttgart, June 1998. `http://www-ci.u-aizu.ac.jp/~herder/publications/ve98-spatial-server/`.

[20] Tosiyasu L. Kunii, Jens Herder, Karol Myszkowski, Oleg Okunev, Galina G. Okuneva, and Masumi Ibusuki. Articulation simulation for an Intelligent Dental Care System. *Displays*, 15(3):181–188, 1994.

[21] William L. Martens and Jens Herder. Perceptual criteria for eliminating reflectors and occluders from the rendering of environmental sound. **J**. **A***cous.* **S***oc.* **A***mer.*, 105(2):979, February 1999. Proc. Joint Meeting of the 137^{th} Regular Meeting of the Acoustical Society of America and the 2^{nd} Convention of the European Acoustics Association: Forum Acusticum; Signal Processing in Acoustics and Psychological and Pysiological Acoustics: Auditory Displays, 1pSP2.

[22] William L. Martens, Jens Herder, and Yoshiki Shiba. A filtering model for efficient rendering of the spatial image of an occluded virtual sound

source. **J**. **A***cous.* **S***oc.* **A***mer.*, 105(2):980, February 1999. Proc. Joint Meeting of the 137^{th} Regular Meeting of the Acoustical Society of America and the 2^{nd} Convention of the European Acoustics Association: Forum Acusticum; Signal Processing in Acoustics and Psychological and Pysiological Acoustics: Auditory Displays, 1pSP7.

[23] Karol Myszkowski, Jens Herder, Tosiyasu L. Kunii, and Masumi Ibusuki. Visualization and analysis of occlusion for human jaws using a functionally generated path. In *IS&T/SPIE Symp. on Electronic Imaging, Visual Data Exploration and Analysis III.* The International Society for Optical Engineering, January 1996.

[24] Karol Myszkowski, Galina Okuneva, Jens Herder, Tosiyasu L. Kunii, and Masumi Ibusuki. Visual simulation of the chewing process for dentistry. In *Visualization & Modelling*, Leeds, December 1995.

[25] Karol Myszkowski, Galina Okuneva, Jens Herder, Tosiyasu L. Kunii, and Masumi Ibusuki. Visual simulation of the chewing process for dentistry. In Rae Earnshaw, John Vince, and Huw Jones, editors, *Visualization & Modeling*, chapter 24, pages 419–438. Academic Press, 24-28 Oval Road, London NW17DX, UK, December 1997. ISBN 0-12-227738-4.

[26] Lothar M. Schmitt, Jens Herder, and Subhash Bhalla. Information retrieval and database architecture for conventional Japanese character dictionaries. In *CT'97 — Second Int. Cognitive Technology Conf.*, pages 200–217. IEEE, IEEE Press, August 1997. ISBN 0-8186-8084-9.