北京交通大學

# 本科毕业设计（论文）

## VR 音频内容编辑系统的设计与实现

## The design and implementation of the VR audio content editing system

学　　院：　XXXX

专　　业：　XXXX

学生姓名：　XXXX

学　　号：　XXXX

指导教师：　XXXX

北京交通大学

2016 年 5 月

# 学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：　　　　　　　　　　指导教师签名：

签字日期：　　　年　　月　　日　　　　签字日期：　　　年　　月　　日

# 中文摘要

**摘要：**虚拟现实是当今飞速发展的科技，电影艺术借助虚拟现实的技术手段已经使其成为越来越受大众追捧的沉浸式体验娱乐之一。本文所涉及的VR音频是虚拟现实电影的重要组成部分，承担提供沉浸式听觉体验的职责。目前的大部分VR电影拍摄使用全景摄像机拍摄然后通过后期图像拼接而成，而录音和配乐则需要通过单独制作，如此一来音频部分和视频部分的衔接就需要重新定位。为了将音频部分对应全景视频中的声源，后期工作人员目前采用手写记录文档，记录音频轨道参数、声源在空间中的坐标和角度、时间信息等，将音频分轨和全景中的声源相匹配。这个操作过程效率很低、准确度有限、体验过程中存在偏差感，后期工作人员急需一种更直观的系统完成这些工作。

为解决这个目前存在的问题，本文中将结合VST架构对VR音频内容编辑系统进行研究。该系统结合Cubase音频工作站，供使用者编辑音源轨道并设置调整VR音频内容。对VR音频与音源方位进行抽象，结合实习公司音频播放引擎提供VR音频的音源展现功能。

该系统的实现提高了音频后期人员编辑VR音频的效率，使音频与全景视频的衔接更加准确。

**关键词：**VR；音频；VST

# ABSTRACT

**ABSTRACT:** Virtual reality is a rapidly developing technology today, the film art with the help of virtual reality technology has become one of most popular immersive experience entertainment. Involved in this paper, three-dimensional surround sound is an important part of VR movies, assume the position of provide immersive audio experience. Most of the current VR movie shooting through the use of panoramic cameras, and then through image mosaic made while recording and music you need to create a separate. As a result, the audio portion and the video portion will need to be repositioned. For the reason of corresponding to the panoramic video source, the staff is currently written record of documentation to record an audio track parameters, the sound source coordinates and angles in space, time information, etc. So it will match the audio track points and panoramic sound source. This operation is inefficient, limited accuracy, and there is a deviation to experience a sense of the process. Staff urgently needed a more convenient system to complete the work.

In order to solve the existing problems. This paper will combine VST architecture to research the audio content of VR editing systems. The system combines Cubase audio workstation for users to edit audio tracks and set the VR audio content. Abstracting VR audio and audio orientation, combined with TwirlingAudio Engine offering VR audio shown function.

Implementation of the system improved the editing efficiency of the staff, so that the audio and video panorama convergence are more accurate.

**KEYWORDS：** VR; Audio; VST

# 目　　录

# 1 引言

## 1.1　项目研究背景

虚拟现实（Virtual Reality, VR）是目前综合性比较强的研究领域，它集多种技术于一身，包括计算机图形学、交互技术、人工智能、分布式、并行处理技术、现代传感技术等，其主要借助这些技术方式配合计算机辅助形成高度仿真的感官体验功能，从而建立一种虚拟现实环境。

虚拟现实已经渗透入我们的生活，目前在多个方面都有广泛的应用，例如医疗、游戏、娱乐、教育、军事等等，虚拟现实技术帮助我们对现实进行模拟，将来的应用前景巨大。其实，直到 1991 年为止，都没有一种商业的 VR 应用，其原因是没有 VR 硬件被开发出来。到 1992 年以后，VR 软硬件逐渐逐渐出现并被许多公司以及个人应用。

虚拟现实强调沉浸感（Immersion）、交互性（Interaction）、构想性（Imagination），这是虚拟现实的三个基本特征，所以虚拟现实主要基于这三个基本特征从视觉、听觉、触觉三个方面进行主要重难点的研究。

在虚拟现实的系统中，听觉是次于视觉的另一个感受传输途径，听觉传输途径传输声音信息到人的听觉系统中，补充视觉信息的单一，增强虚拟现实的沉浸体验感，比如播放三维场景的虚拟空间中同时播放适当的声音，可以使体验者更清晰的体验到虚拟现实场景中的物体动态，也可以减轻大脑对视觉的过度依赖性，降低沉浸式虚拟现实对视觉信息的要求，同时从虚拟现实环境中获得的信息更多了。

VR 音效帮助人耳对声音进行定位，人们可以依靠声音辨别音源发出的方位，并且依据声音判断环境等。VR 音效在实际中也有非常重要的应用，在电影领域，理论上需要双耳效应利用耳机才可以实现一个准确的三维音景，巴可公司在 2013 年研制出了革命性的 Auro11.1 三维音效技术，使除却耳机后任可享受三维音频的电影，这项技术让观众享受到了最自然的立体声电影。

## 1.2　项目来源

VR 电影的制作，时至今日，已经是围绕着其内容本身来制作为主，愈来愈多的工程资源可以使用，已经不是很多年前开发动画的样子。团队内容制作的部分主要分为后期部和技术部，技术部负责开发 VR 内容制作所需要的技术和工具，而后期部则主要通过这些开发的 VR 工具或者现有技术进行 VR 电影的内容制作。VR 电影制作的软硬件包括 360 声场录制机、全景摄像机、VR 全景声引擎、多路视频拼接、VR 移动端 app

等。公司具有这些软硬件的知识产权，属于自主研发的VR工具，在电影开始拍摄的过程中使用全景摄像机对视频进行采集，并使用360声场录制机进行音频的录制，这样VR电影的基本素材就有了，包括一个完整的未进行拼接的全景画面素材以及多轨道音频。全景画面通过多个摄像机同时拍摄多个角度，得到素材后通过图像拼接技术拼接画面为全景画面。音频的录制则需要多步骤的加工，其原因是环境音效的复杂性，当一个VR电影在拍摄时，360声场录制机录制得到的音频是真实实时的环境音数据，而这就对拍摄环境的要求极其苛刻，否则无法达到电影高清音效的要求。这也是低成本VR电影所遇到的问题之一，360声场录制机录制的效果反而不如单独制作高清音效。

独立录制音效后，VR电影中的音效会清晰很多，音源位置变得准确。准确的人物配音、环境物体的特殊音效、环境背景音乐等等，这些被独立制作后相互分开，不再是整体的立体声环绕声，而是定位于固定方位的不同音效。相比用360声场录制机录制的VR全景声，单独制作的VR全景声赋予了VR电影全新的沉浸体验感。让观众在使用VR头盔观看全景视频的同时，VR音效也可以同时帮助观众定位VR电影中发生的一切。

独立制作VR音效，意味着VR电影中的每一处发声的物体都要被单独录音，人物声、液体声、脚步声，甚至特殊的噪音都要单独制作，为了给观众身临其境的感觉。独立制作VR音效，会产生多个音轨。每一种类的音频都会独占一音轨，VR电影中的环境越复杂，意味着需要的音效就会越复杂，所以常常音频后期人员会面临巨大的工作量。在后期人员的工作当中，除了创作制作音效外还包含一部分工作就是在视频中定位这些音频，确定VR电影环境和VR音效的匹配，保证VR电影体验的质量达到应有的沉浸感和准确度。所以，后期人员在制作音效后，还需制作该音效在电影场景中的元数据，这些元数据保证了音源发生的方位、时间、强度等信息。这样的工作流程决定了当VR电影不符合要求的时候，会面临巨大的修改，而且在目前元数据无法保证准确性，全部由后期工作者靠人眼调整和对位。多个音频数据需要更高效的被编辑修改，从而提高VR电影制作的效率和质量。

当全部的元数据整合完成，一部VR电影的音频素材就已经完成，可以作为音频数据由VR电影播放器进行播放。下面介绍论文主要工作。

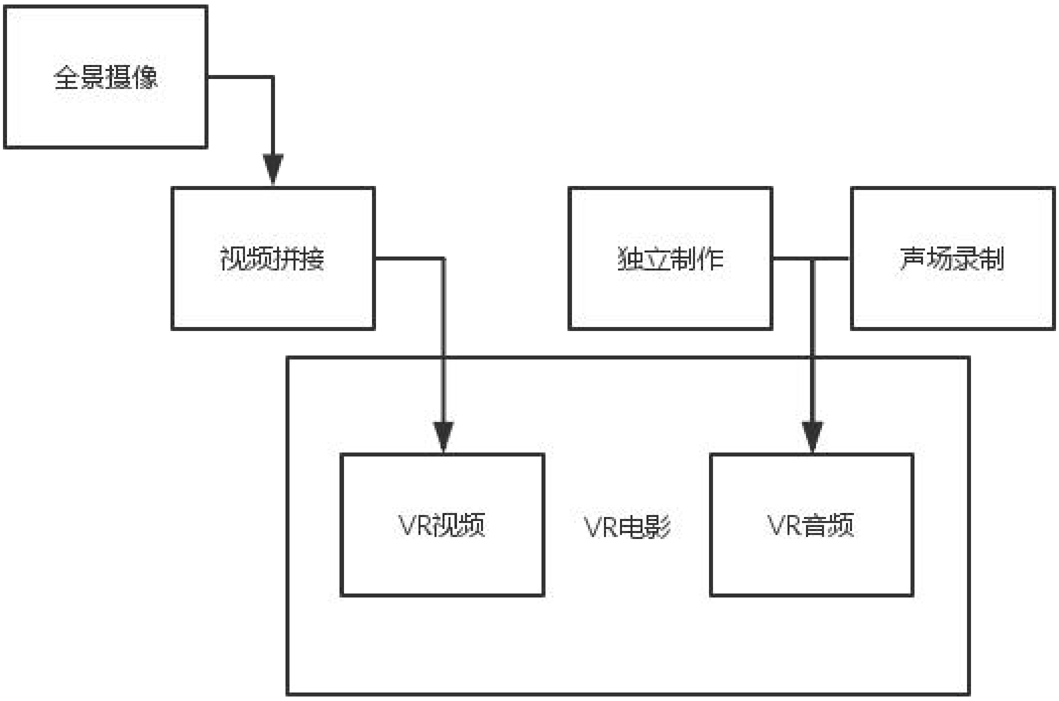


图 1-1 低成本 VR 电影制作流程示意图

## 1.3 论文主要工作

本人主要参于该系统的设计开发工作，在实习期间的前期学习了 VST 框架以及熟悉作为宿主软件的 Cubase 音频编辑系统，后期配合主工程师学习并开发 TwirlingAudio 的音频模块等。

本文主要基于在实习期间的经历，前期大量的查阅相关资料，VST 框架并不为人所熟知，开发者比较少，所以相关文献也很少，花费了很多精力和时间做前期准备工作，对于后期工作人员的工作流程也进行了很细致的了解，并跟随制作组参与 VR 电影的拍摄，了解 VR 音效原创工作。

该 VR 音频内容编辑系统的设计过程中，主要参与了项目的需求分析、系统设计以及实现等部分，通过对 VR 音频制作的过程进行分析，明晰目前音频制作过程存在的问题，配合后期工作人员设计相关功能。

首先，通过对 VR 音频内容制作的需求分析，将该系统分为三个主要工作模块，即音频效果编辑模块、音频元数据编辑模块、音源展现模块。音频效果编辑模块主要借助 VST 框架完成，借助 Cubase 对音频的内容提供调整、制作的功能。音频元数据编辑模

块提供用户编辑音频元数据的功能。音源展现模块借助 TwirlingAudio 完成，测试音源在空间方位中的展现。

这篇论文主要阐述了该项目分析和设计的过程。

## 1.4 论文组织结构

本文主要分为六个章节，其主要内容是以下章节。

第一章：引言。主要讲述项目背景以及项目来源，结合相关资料以及实习中学习到的内容，从虚拟现实的背景到 VR 电影的创作流程，整体的对项目来源进行了介绍，并对论文的相关信息做了简单的描述。

第二章：介绍项目中所用的到相关技术理论。从技术途径选择的原因到技术内容的解释，帮助阐述选择的技术路线。

第三章：项目的需求分析，从解决需求问题的途径进行阐述，描述用户功能性需求和非功能性需求。

第四章：VR 音频编辑工作流程中设计逻辑框架，对 VST 架构进行相关研究并设计系统架构。

第五章：介绍数据库的设计与实现，从概念模型到逻辑模型和物理模型进行数据模型的建模。

第六章：阐述 VR 音频内容编辑系统的详细设计，主要的三大功能模块：音频效果编辑模块、元数据编辑模块、音源展现模块的类设计。

第七章：为结论与总结，主要对这次项目的工作进行总结，并且表达一些对此次工作的深思，展望对今后的相关领域的进一步研究。

# 2 项目相关理论和技术基础

音频内容编辑系统是基于 VST 架构的开发项目，通过 VST 框架以及 Cubase 的相关接口实现。音源的展现通过 TwirlingAudio 实现。

## 2.1 VR 音频内容编辑系统的理论基础

VR 音频在电影中想要与视频相互接轨，需要相应的对接方式，采用元数据编辑的方式确定 VR 音频在电影中的方位可以确保准确的将 VR 音频与视频衔接。VR 音频元数据主要包括空间位置、时间、音轨等信息。在项目开始前，需要对 VR 音频的影响因素进行技术和理论的认识。

### 2.1.1 VR 音效影响因素

1）方向

人类耳部的构造非常的复杂，能捕捉到很微小的声音变化，在人们的日常生活中声音的传播常常具有方向。所以，当收听的人站在一点不动而接收来自四面八方的声音时，会在心理和生理上产生不同的效果。而通过研究表面，人能够感应声音的方向，其原理是通过耳鼓之间的电平差。在这个坐标系内，决定听者的因素有两个，一个是前向向量，另一个是顶向向量。

声音在传播的过程中具有传播方向以及传播速度，这两个因素决定了声音的方位。传播的方向又分为定向传播和非定向传播两种。而定向传播则可以在传播方向上产生比较大的影响，因为假定在一个空旷的没有遮罩的空间中，音源部分发出的声音会向周围扩散，但是在每一个方向上的振幅相同，所以定向传播比非定向传播要产生更大的效果。除此以外，声音在各种介质中的传播速度不一样，传播速度和介质的密度成正比，同时也会随着介质的环境属性的变化而变化，比如温度、压强的改变。

2）距离

三维音效技术非常注重距离对于音效的影响，音源距离的变化会给听者带来整体上的听觉变化。距离主要在三个方面影响 VR 音效的效果：首先，两耳之间的距离会影响听者判断音源的前后方位，双耳效应会给听者不同的听觉感受；其次，音源距离听者的距离也是重要因素，通常情况下，距离影响音量的大小是最主要的，距离远则音量小，距离近则音量大，当然还会影响音源的广度。在程序设计中，应该考虑最大距离和最小距离，这样才可以控制音源距离关于音量的改变，不至于音量消失或者音量过大；第三，衰减参数也是影响因素，距离不同、声音不同具有不同的衰减参数。

3）运动

多普勒效应所影响的音源和接受声音者运动所产生的声效改变。

4）环境

在 VR 环境中的考虑，不同的环境对声音有不同的影响，听者产生生理效应也不同。比如介质对声音的反射和传播效果会不同，这些属于次要问题，但是对一个极其准确的音效模拟来说却必不可少，所以需要根据更加准确的数据进行具体的考量。

### 2.1.2 人耳定位原理

通常情况下，人通过双耳效应定位音源位置，但是这是一种模糊的定位，因为在现实中，人们可以很准确的定位声音发出的方位，人会根据声音的发出位置、传播的方向、距离等等，然后依据周围的环境，在大脑中构建空间感。然后，依靠视觉、听觉、感触等感官的整体感受作用准确感知现实的世界。这是一种复杂的作用体系，它们相互配合相互纠错，丰富的感受周围的环境，最后我们才可以明确声音准确的方位，并且感受到声音的距离以及空间感。

这里涉及到一种听觉心理。简单的来说，就是人脑解释声音的方式。听觉心理除了主观感受到声音的位置、方向、距离三种属性以外，还包括遮蔽、高频定位、余音等。音源在发出声波经过传播后，最后到达脑部，在这种传播过程中，会产生两种差值，为双耳时间差（ITD）以及双耳声级差（ILD），然后结合之前的听觉系统分析，我们可以感知声音的方向、运动、距离等信息。在这里不做过多介绍，因为本系统不会涉及，该理论主要应用在产生 VR 音效播放效果上，应用到头部相关传递函数（HRTF），以此来产生准确的自然声场，属于三维音效效果表现部分。

## 2.2 VR 音频内容编辑系统的技术基础

系统的音频编辑功能通过VST框架实现，音源的展示模块使用TwirlingAudio实现，在这里浅析 VST 架构的主要组成，其中包括 VST 架构所使用到的主要技术，以及音频模块的主要元件。

### 2.2.1 VST 架构

大约二十年前，如果音乐家被告知经典合成器成功地被以软件形式重新创建，并以原价的一小部分出售，他们会不约而同的笑。他们不太相信自己大约有一半的人或更多的人会融入这个创作形式，就因为深受音乐工作者喜爱的 MIDI 以及音频音序器将会出现。然而，VST 架构正是这样一个新产品。

VST 系统是由斯坦伯格公司研发，它将一个完整的工作室通过软件的方式进行实现。事实上，在其最早的版本，它是允许被第三方开发人员开发并能"插入"到宿主应用程序的实时效果模块。斯坦伯格公司推出了标准的 VST 插件的第二个版本，它可以发送 MIDI 数据，并对这些数据进行修改。这使开发人员可以添加更多的功能，例如 MIDI 参数控制的效果和锁定的节奏的效果设置。VST 架构进步的必然结果就是之后的 MIDI 信息也可以被用来运行 Synth 引擎，而不是只是影响处理器。这些合成器伪装成效果器插件并被适用于相关的使用环境，被称为 VST 文件或 VSTi。VST3 架构处理简单对象流程如图 2-1 所示。

VST 效果处理器或文件的所有函数都通过硬件控制器直接可控和自动化。VST 可以很容易集成外部设备，允许度身订造。

VST 技术已完全改变了世界的音乐制作。有的插件和工具，可能永远不会存在于物理世界中，但有提供独特的功能和提供这种技术的可能性。随着不断增长的开发人员和增量技术的进步，可以假定未来的技术只会更好。

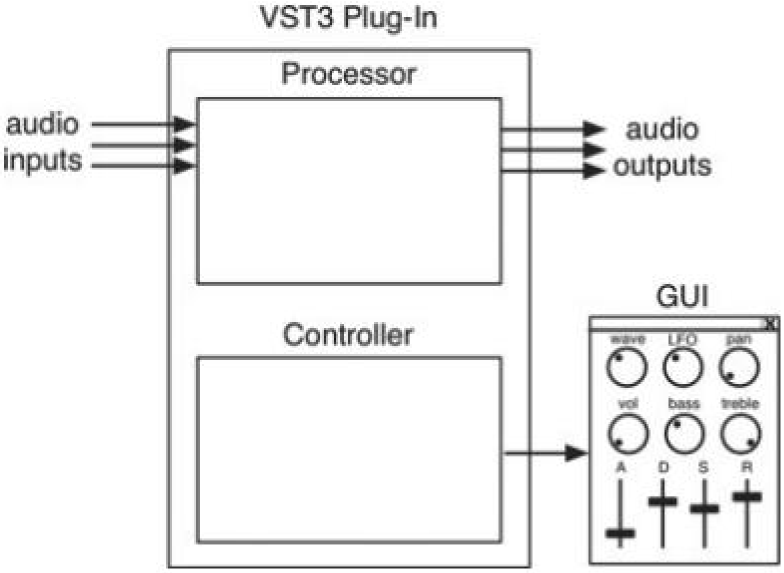


图 2-1 VST 框架处理流程图

在项目中，采用的是 VST3 架构，相比 VST2 架构，VST3 架构比老的 VST 2.4 和 VST 2.3 强在有更少的 CPU 占用，在读取出来并不使用的情况下不会再占用 CPU 资源；还有更多的动态 I/O，任何插件都可以调用单声道、立体声或环绕声（5.1）的通道，也就是说任何 VST3 插件都支持环绕声。它有可自定义通道数，VST3 插件允许你手动关掉不用的通道。还有更灵活的通路，VST3 插件可以自己设置内部通路，比如一个带声码器的合成器插件，可以直接将一条音频通道接到它里面作为效果器用。它支持 side-chain 侧链并且支持多 MIDI 输入端口，VST3 插件可以同时有多个 MIDI 输入端口，也就是说打开一个插件，就以用多台 MIDI 键盘去演奏。它采用树型结构的参数，所有老 VST 的控制参数都是同级的，VST3 则可以有树型结构，比如 A 参数下面展开 A1、A2、A3 多个参数。还有重要的一点，它可调节的窗口大小，而老版本 VST 的窗口都是固定大小。

### 2.2.2 组件对象模型（The Component Object Model）

VST3 通过组件对象模型或者说 COM 进行开发。COM 是微软的发明物。简单的来说，组件对象模型的编程就是一种开发软件组件的方式。它定义了相关对象在单个应用程序内部或者多个应用程序之间的交互行为方式。COM 是 Microsoft 早在 1993 年便提出的组件式的软件平台，用来协助开发进程间的通信以及被当作组件式软件开发的有效平台。COM 其实提供跟编程语言无关的方法来实现一个软件对象，因此它可以在任何其他环境中运行。COM 要求其软件组件都必须遵照一个共同的接口，该接口与具体实

现无关，因此可以隐藏相关的实现属性，并且会被其他对象在不知道其内部真实实现的情形下正确的使用。

VST 架构 Controller 组件实现了 COM，文件系列化，插件初始化后可处理音频和处理 MIDI 音符事件并接收控制变更信息。

以下是 VST 的 Controller 创建主要方法：

1）COM 的创建：createInstance()；FUID cid；

2）初始化：initialize() ；terminate()；

3）连载：setComponentState() ；setState()；

4）设置 MIDI 控制器分配：getMIDIControllerAssignment（）；

5）创建一个自定义视图：CreateView（）；createSubController（）；

6）处理核心图形用户界面和参数的功能：EditController（）；

7）在 MIDI 映射接口：IMidiMapping（）；

8)处理自定义视图创建和接口操纵杆和其他更复杂的控制：VST3EditorDelegate（）。

### 2.2.3 插件（Plug-ins）

该插件核心是你需要实现与客户沟通，建立一个渲染系统和处理的图形用户界面（GUI 或 UI）的详细信息，交易代码。虽然实现细节有很大的不同，操作的基本理论实际上和任何其他插件格式相同。

### 2.2.4 动态链接库（DLL）

要构建应用程序被编译并卖给顾客需要不同的策略扩展应用程序的功能组件。解决的办法是在运行时链接到功能。这意味着这些预编译的功能将在一个单独的文件中，该客户端将被链接，但只有在开始运行后存在沟通。这种连接被称为动态链接或显式连接。包含新的函数的文件称为动态链接库或 DLL。在 Windows 中，该文件通常使用扩展名。然而 DLL 在 VST3 中重命名扩展名为 .VST3。

为了使用该代码在 DLL 中的客户端必须执行两项活动：

1.加载该 DLL 到进程的地址空间；

2.建立沟通机制，从 DLL 中调用的函数。

### 2.2.5 处理器和控制器（Processor and Controller）

VST3 使用了两个独立的 C ++对象来分别处理音频流程和用户交互的两个主要任务。这些被命名为处理器和控制器对象。

Processor 对象将从一个名为 AudioEffect 的基类那里继承，这个对象将处理音频信号，这里合成也称为渲染工作。这将 ALS 接收控制变更信息，当用户调节控制，并根据需要将修改处理。该 Processor 对象将回答来自客户关于音频布辛能力和音频格式的能力查询。它也将接收，解码和并注意 offmessages 处理 MIDI 音符。处理器还实现序列的任务，这意味着在任何给定的时间，串行二进制文件进行装载以及进行插件的状态的存储。术语连载被使用，因为这些参数将被存储在系列中。它允许客户端来初始化你的插件时，预留该项目上次的保存/关闭，以及提供很基本的用户预设加载和存储能力。

控制器对象将继承称为 EditController 的基类。在 VST 架构里，术语“edit”和“editor”是指 GUI 或与其相关的对象，而不是一个文本或音频编辑器。处理器已经接收控制变更信息，但是还需要一个单独的控制器对象。控制器对象将处理 GUI 控制参数的初始化和设置并执行发送和接收信息，并从 GUI 的通信机制。这使得处理这一改变声音的渲染控制变更消息进行单独的作业。如果用户希望记录和播放控制动作的控制器还可以处理你控制的自动化。该控制器还必须处理序列化，但只有在读取端。最后，控制器对象也将涉及 MIDI 控制器的设置。

处理器：

1. 初始化和从主机大约通道数量和音频格式处理 GUI 控制变更查询；
2. 响应 MIDI 音符，并留意关事件处理（渲染）音频流；
3. 插件的参数和从文件的序列化（读/写）；

控制器：

1. 声明和初始化的 GUI 控制参数；
2. 实现发送和接收参数的改变 MIDI 控制器设置；
3. 读取端序列；
4. 创建和自定义 GUI 的维护。

### 2.2.6 Cubase 音频编辑 DAW 系统

Cubase 音频编辑系统允许操作者编辑 MIDI 文件，原始音轨，和类似的歌词等相关信息，以及范围内的格式，包括乐谱，编辑控制台，事件列表，以提供它们等操作员也可以混合不同的轨道成立体声.wav 文件准备红皮书格式刻录到光盘，以及.mp3 刻录到 CD 或 DVD 的文件，或者要在 Web 上发布。

Cubase 在 1999 年推出了被称为的 VSTi 软件合成一个虚拟仪器接口。这使得 Cubase 有可能为第三方软件程序员创建和销售虚拟仪器的平台。这种技术已经成为其他 DAW 软件的标准，在 Macintosh 和 Windows 平台上集成基于软件的乐器。

Cubase 可以非常好的兼容 VST 插件进行编辑运作，可以说随着 VST 的发展，VST

就是 Cubase 的主要编辑模块。

### 2.2.7 音频数据波形文件

音频数据文件的格式主要分为两大主类，波形文件以及 MIDI 文件。波形文件就是我们常使用到的.wav、.MP3、.asf 等格式的文件。它们主要记录音频声波的特征值。而 MIDI 文件是数字音乐数据文件，它将音乐的弹奏过程记录下来，比如什么乐曲的什么音调、力度多大、用时多长等信息，所以在数字音乐创作中常使用 MIDI 格式的音乐文件。

### 2.2.8 TwirlingAudio

公司专利的 TwirlingVR 全景声引擎提供高标准的耳机三维全景声解决方案。利用独有的全景声场模拟算法，将环境声和虚拟声源精确的还原在三维空间里。声源的位置，方向，距离以及运动轨迹等都能够通过耳机被清晰和准确的感知。特有的声场空间变换算法，确保多声源复杂场景能高效准确的呈现。特别适合对运算复杂度有较高要求的 mobile 应用。支持头部旋转和移动下的声场重建。跨平台引擎，可以灵活的用于 android，ios，windows，Mac OS 以及 unity 平台。

### 2.2.9 音频剪辑（Audio Clip）

任何被导入的音频文件都被称为音频剪辑（在这里简称音频）。音频剪辑工作是与音频源和音频侦听器结合使用，只将剪辑作为音频数据。比如当你在场景中将剪辑添加到一个对象上，它变成一个音频源，并且现在有音量，音调，以及循环这样的其他属性。一个源播放时，音频监听器可以监听到所有距离范围之内的源，然后通过扬声器让你听到。显然在场景中只能有一个音频监听器，而它一般是添加在主摄像头上。在大多数情况下对于音频源进行操作，将多于音频剪辑和监听器。

音频剪辑简单来说就是被音源所使用的音频片段。3D 声源会被模拟在三维场景中当做环绕音效进行播放，所以这对于系统实现来说，使用音频剪辑进行多个轨道的播放非常方便。 3D 声音会凭借衰减音量以及扬声器之间平衡调整来模拟声音的距离感以及方位感。单声道和多声道的声音数据几乎都可以放置在整体的三维空间中。

### 2.2.10音频源（Audio Source）

音频源可以在三维场景中播放相对应的音频剪辑。如果恰好音频剪辑是一个 3D 音频片段，而音频源是在一个给定的位置，那么音频源就会随距离衰减这样的方式进行播

放。音频既可以在扬声器之间传播也可以在 3D 和 2D 之间进行转换。

有音频的几个的属性，作为音频源和音频侦听器之间的距离的影响参数。

1）音量：幅度（0.0-1.0）随距离改变

2）平衡调整： 左(-1.0)到右(1.0)随距离改变

3）扩散：角度(0.0 到 360.0 度)随距离改变

4）低通：低通截止频率（10.0～22000.0hz)随距离改变。

# 3 项目需求分析

项目的需求来源为公司内部的后期工作者，他们扮演用户角色，事实上，正是这些用户所遇到的问题需要被解决。在进入团队后，首先，发现主要目标用户，对目标用户的工作流程进行了解，从而获取真实的用户需求。大多数需求问题通过反复的沟通与了解，逐渐清晰，再通过作者本身的实时考察，跟随用户真实项目的工作流程，分析与研究后，得出相关结论。在这个过程中，渐渐加深了对 VR 电影的制作流程的认识，以及 VR 音频内容制作上的需求了解，以下主要介绍 VR 音频内容编辑系统的需求分析从问题映射到需求的过程。

## 3.1 项目系统描述和功能性需求

VR 音频内容编辑系统本质上为了提高后期人员工作效率，提高 VR 音频元数据的准确性，所以设计该系统的过程中，会主要以这两点主要目的为需求导向，分析系统的主要功能模块。

VR 音频内容编辑系统的主要功能模块分为三大模块：

（1）音频效果编辑模块。音频效果编辑主要包括音频增益效果、延迟效果、单声道转换立体声功能；

（2）音频元数据编辑模块。提供用户编辑和导出音频元数据的功能，元数据的详细内容包括方向、距离、时间、坐标位置的信息，这些模块帮助用户编辑音频在空间中的音源位置、声音属性和在 VR 电影中的播放时间。

（3）音源展现模块。此模块展现音源在固定位置播放音频的功能，主要为展现全景空间中音源播放音频的位置，从而帮助用户预览和更准确的定位音源在视频中的位置。

### 3.1.1 音频效果编辑模块功能性需求分析

用户通过音频效果编辑模块编辑音频主要内容，音频主要内容涉及音频效果、音频

的声道格式、元数据的选择等等。VR 音频内容在独立制作后再在此模块进行相关内容的编辑操作。

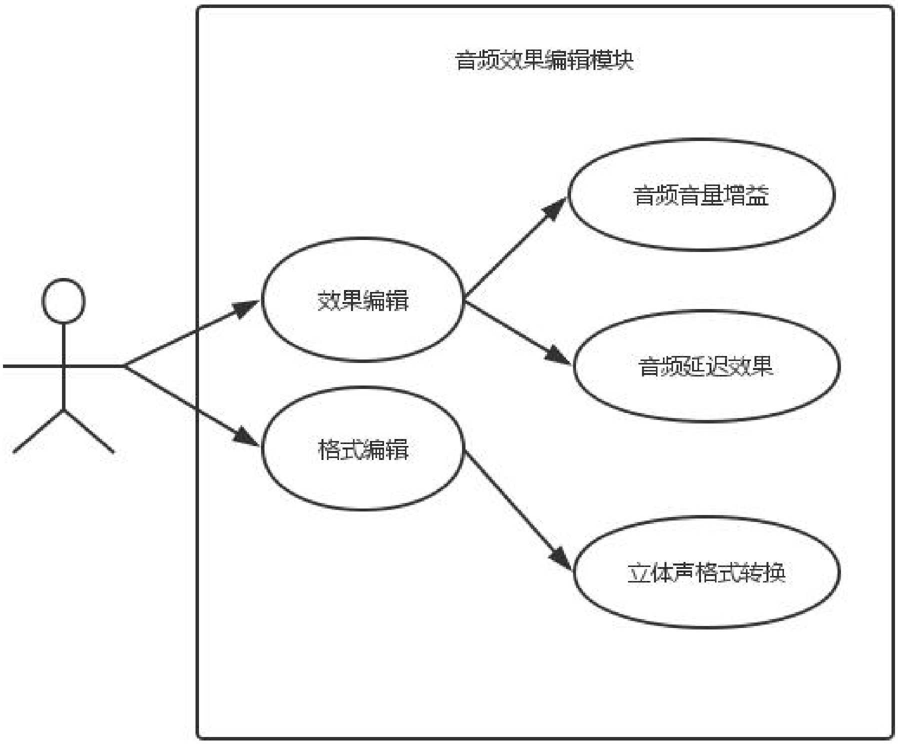


图 3-1 音频效果编辑模块

该模块包括以下的功能需求点：

（1）效果编辑。后期工作人员常常需要一些简单的效果来控制音频的输出，当需要根据距离的改变而改变音频的输出效果的时候，我们常常需要给音频添加增益、延迟等效果器。这一点来看，该模块需要几个常用的效果器功能。

音频增益（Gain）效果：改变音频的输出流，控制音频的输出流，从而控制音量的大小。增益效果无法为负数，因为增益效果常常用来加强音频输出，最低为 0，当增益为 0 时，音频播放音量为 0，意义为静音。

音频延迟（Delay）效果：音频延迟效果使音频数据增长，是一种常用的效果器效果，延迟效果可以用来将音频数据拉长，是播放速率减缓，有效的增大音频的细节部分，在编辑的工作中是非常常用的效果。

（2）格式编辑。一般来说，导入的音频文件只生产一个轨道，但这种情况下，存在的单声道音频很难与其他的声道的音频相互配合，立体声为双声道格式，当用户使用到双声道格式的音频，在 VR 电影中是无法符合要求的，在 VR 电影中，多个声道的音频，在分别制作之前必须为单声道格式，否则无法进行多声道合成。立体声与单声道的

格式转换功能可以通过 VST 架构很简单的实现。

### 3.1.2 元数据编辑模块

元数据承担音频内容与 VR 空间的衔接功能，是该 VR 音频内容编辑系统的主要产出物，用户在 DAW 系统中导入音频并编辑，为了后期衔接 VR 视频，还需要对此音频数据的元数据进行编辑，才可以保证后期音频与 VR 空间中的音源相比配。元数据的内容主要体现音频音源的相关位置、时间、轨道参数。元数据编辑模块的用例如图 3-3 所示。

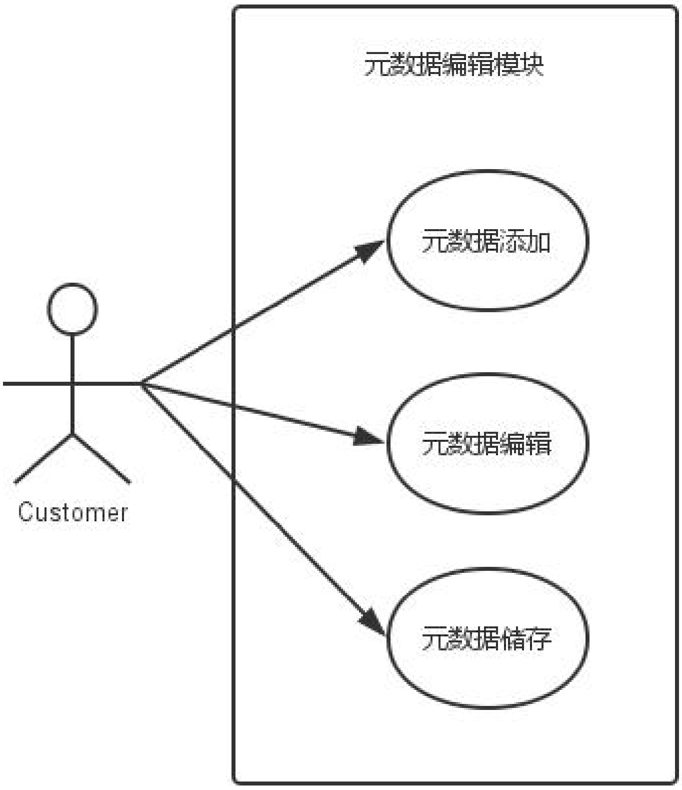


图 3-2 元数据编辑模块用例图

元数据编辑模块保证用户可以对音频元数据进行添加、编辑加工和储存。用户在开始前，需要给音频添加 Metadata，系统相应用户创建新 Metadata 文件，通过用户在系统上的编辑修改后保存。

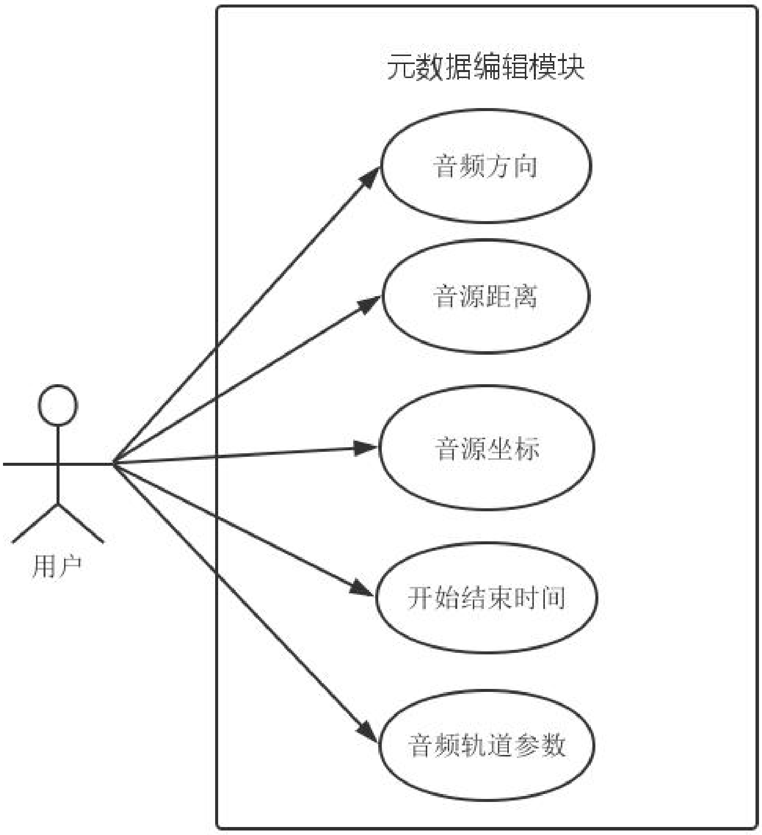


图 3-3 元数据编辑模块

元数据编辑模块包括以下具体功能：

（1）音频方向编辑。音源在初始化时对于摄像头，也就是听者来说具有方向角度，包括水平角度和垂直角度，用户在元数据中需要编辑该参数用于初始化音源方向；

（2）音源距离编辑。用户需要根据音源与摄像头也就是听者的实时距离编辑每个时间段的距离变化，这个变化与时间参数相关联，用户通过记录时间节点和距离的方式来记录这些变化；

（3）音源坐标编辑。用户首先需要在相应的时间节点确定音源的世界坐标，与音源的距离变化相同，用户通过记录时间节点和坐标的方式来记录这些变化；

（4）开始结束时间。VR 电影中音效的出现结束时间也可以在这里进行编辑，负责音量的大小设置、相关的起始结束时间；

（5）音频轨道参数。元数据的编辑首先需要设置音频轨道，确保编辑了正确的音轨元数据。

通过以上的参数，用户通过该模块调节这些参数，设置出符合标准的元数据，并导出元数据，元数据存入数据库，与音频相互关联，便于公司内容对音频内容进行管理，元数据需要多次被修改，所以通过 VST 接口实现的编辑模块可以符合实时编辑元数据

进行保存这个功能。

遗憾的是使用 VST 架构的模块无法与公司音视频的 SDK 相结合，无法直接将元数据展现在 VR 视频中，只能通过存入数据库，保存在本地等方法再与音频同时导入 SDK 进行解析播放。

### 3.1.3 音源展现模块

音源展现模块提供用户预览音源播放音频效果的功能，用户在编辑元数据后需要参考音源播放音频表现判定音源位置是否正确，通过该模块虚拟环绕播放，使音源播放音频体验感类似空间中的固定位置。用户点击预览模块，选择导入 metadata，并播放 metadata 相对的音频数据。这个过程一般在编辑元数据之后，用户使用该功模块需要已有元数据和音频数据，否则无法使用。音源展现模块的用例图如图 3-4 所示。

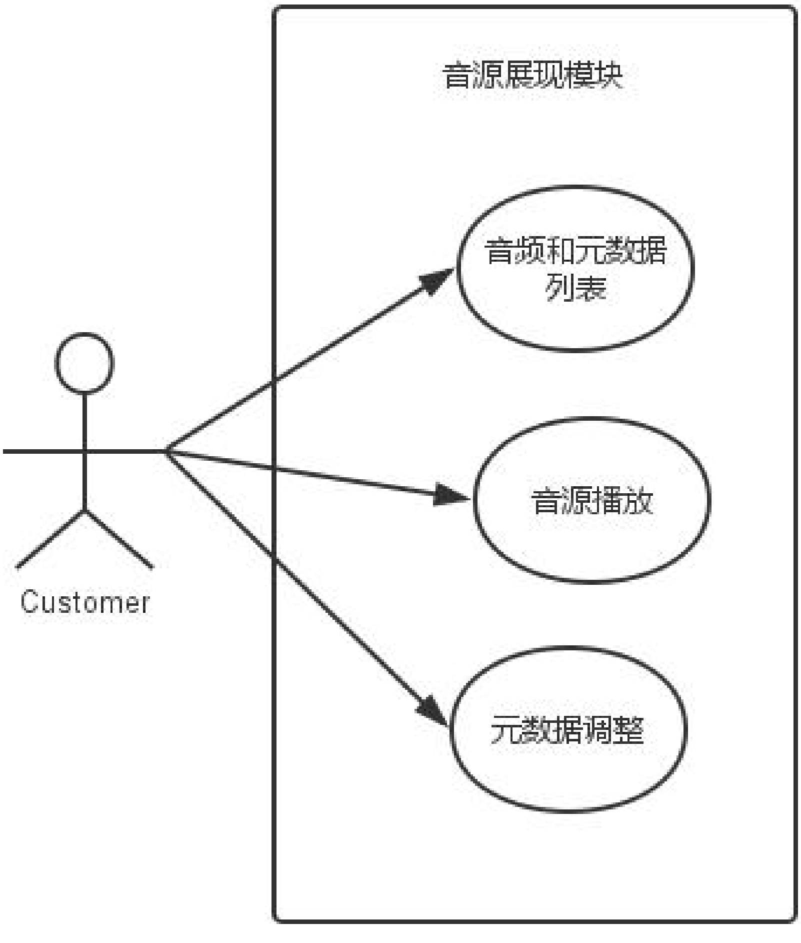


图 3-4 音源展现模块

该模块包括以下功能：

（1）音频和元数据列表显示

用户通过音频和元数据列表选择进行音源展现的音频或元数据。

（2）音源播放

具备播放音频的功能，可以听到虚拟音源的环绕声，从而模拟 VR 电影中的音源位置。

（3）元数据调整

这里同时也需要元数据编辑功能，用户可以通过该模块测试音频元数据是否准确，调整音频元数据，保存元数据。

## 3.2 项目非功能性需求

VR 音频内容编辑系统主要达到的功能需求是可编辑元数据，音频内容与视频内容更好的结合需要提供工具给用户进行准确的定位，用户在使用系统进行音频导入、编辑、预览的过程中，必须达到可以提供用户完成这一工作流程的使命。用户在使用该系统的过程中，需要更加便捷的简单的功能，显而易见的窗口和界面，提高工作效率。

### 3.2.1 可靠性

音频编辑的数据文件属于业务中的主要产品，丢失将对公司造成很大损失，并且延缓开发进度，所以系统在工作过程中，需要良好的可靠性。

音频内容在用户操作的过程中不能丢失，产生的错误应该可恢复，可靠的系统应该考虑到系统奔溃或者异常产生的严重后果，所以应该保证系统在用户操作过程中的稳定性，并且不会对音频数据造成不可逆的改变操作。

### 3.2.2 易用性

系统主要供后期人员使用，音频后期人员更熟悉效果器的传统音乐元件的交互界面，所以在界面的制作上，需要更多的非技术性考虑。通常的效果器比较丰富，界面花哨，而且符合音乐工作者的使用习惯。

### 3.2.3 效率

本系统达到的主要目的之一就是提高工作人员的编辑效率。所以在用户使用该系统

的过程中，需要快速的编辑过程，过于复杂的编辑过程会导致软件的是使用必要性不高。这是一点主要的非功能性需求，所以尽量的简化用户使用方式。

用户在使用音频效果编辑功能时，需求为快速编辑调整音频为 VR 音频，不需要再对音频做相关的内容修改，主要的需求是对元数据进行编辑，不是对音频进行创作。这一点需要明确。

## 3.3 本章小结

本章主要从系统的主要描述、功能性需求分析和非功能性需求两大部分介绍了目前用户需求，功能性需求主要从系统的三大模块进行介绍，音频效果编辑模块的需求主要从用户常用角度出发进行考量，元数据编辑模块主要从系统主要功能需求的角度进行思考，音源展现模块主要从现在可行的实现方案方面进行挖掘。非功能性需求从用户实际需求角度出发，考虑系统可靠性、易用性和效率三个主要方面。

# 4 系统架构设计

从了解到的功能性需求以及非功能性需求来设计整个系统。作者在开发过程中遵循面向对象设计原则，提高开发的实际效率和代码的可靠性。

整个系统首先基于 VST 架构设计开发音频效果编辑模块和元数据编辑模块，后结合 TwirlingAudio 开发音源展示模块，配合数据库对音频、元数据进行数据管理，增强了系统的易用性和效率。

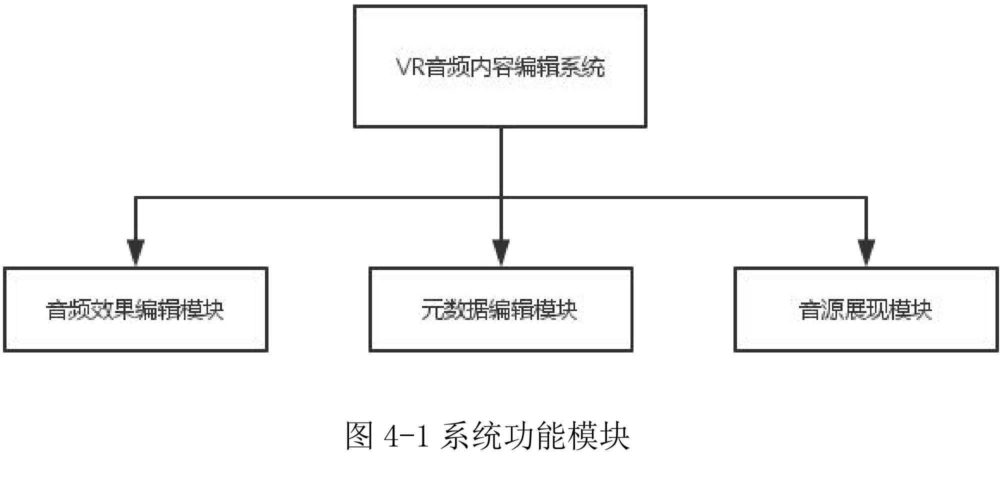


图 4-1 系统功能模块

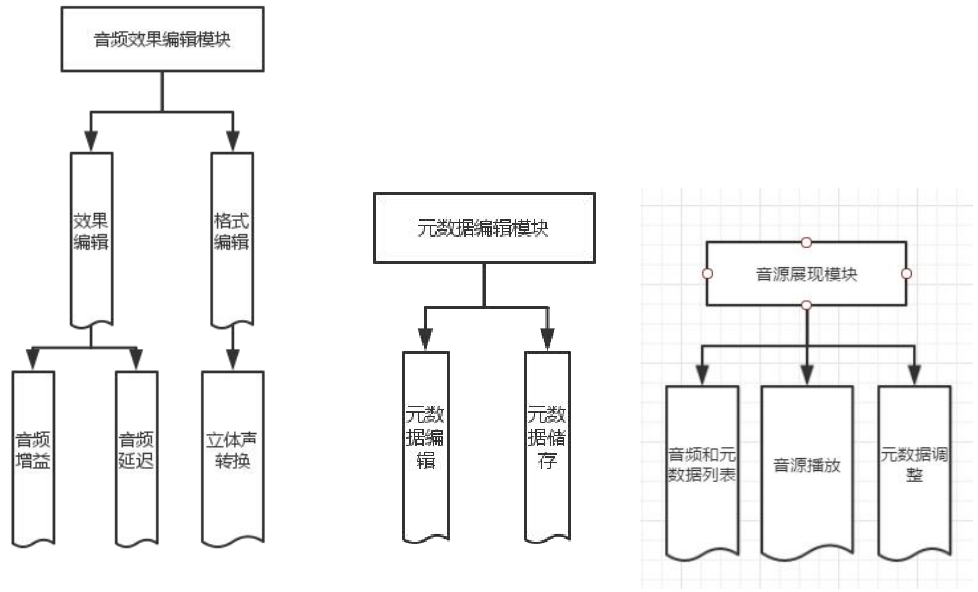


图 4-2 子模块功能

## 4.1　VST 架构分析与设计

### 4.1.1 系统 VST 架构分析

图 3-7 示出了 VST 插件的结构。处理器和控制器的接口由圆和线表示。图 3-8 是 VST 处理器架构图。

在处理器方面，IAudioProcessor 接口公开更多的接口，可以用它来接收 MIDI 事件，控制信息和音频 I / O,尽管插件将只输出。在控制器 IEditController 保持与绘制 MIDI CC 的 GUI 参数，IMIDIMapping 以及 IPlugView 用于实例化 GUI。

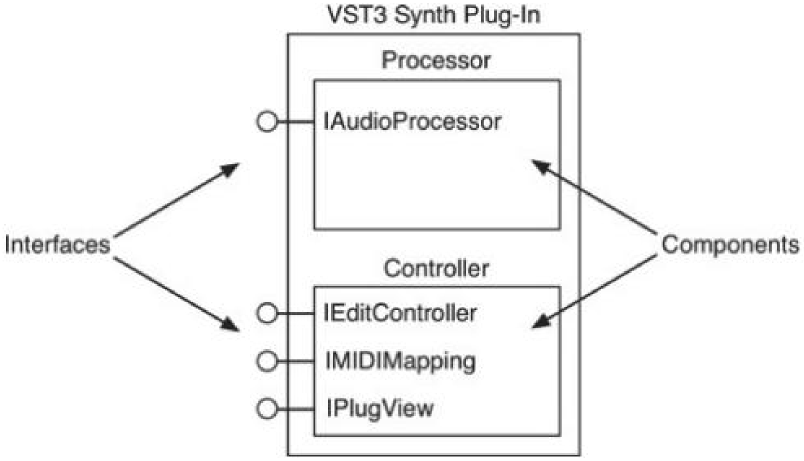


图 4-3VST 架构图

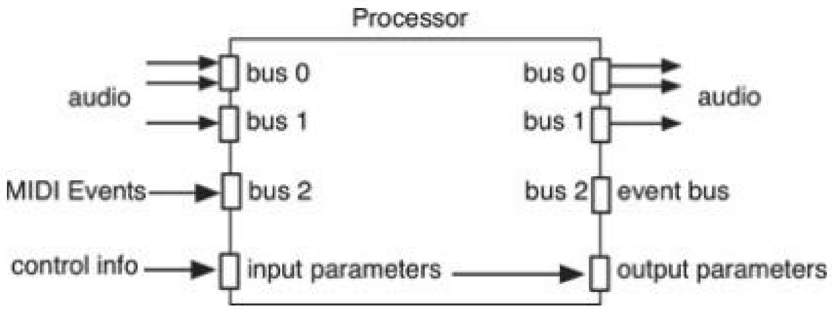


图 4-4Processor 结构图

音频 wave 和 MIDI 事件都需要占用 bus 接口，在图中命名 bus0、1。VST 模块可以具有输入和输出 bus 的任意数量组合。在最通用的形式中，处理器 Processor 可以接收和传送 MIDI 事件和控制 wave 音频数据变化信息。

音频 bus 可以有任意数量的 channel。合成插件要执行总线 0 的单一的音频输出 bus，该 bus 是立体声的两个通道。 MIDI 事件 bus 支持 MIDI 通道，插件将只接受输入参数，传递参数和 MIDI 信息出来的进行试验。

Controller 控制器组件描绘了 GUI 如何与它的容器参数进行通信。GUI 可以定制设计，继承 IEditController 实现 Controller 的自行设计。 MIDI 数据可以被映射到 GUI 控件并且作为外部控制机制。MIDI 数据在此我们不做讨论，本系统的主要音频数据对象为 wave 波形音频数据。此外，还可以使用子控制器作为助手，做一些比较复杂的控制，控制器对象操控使用不必声明任何 bus，这些都是通过 VST 框架底层接口做的工作，所以实现控制器组件和主要 GUI 界面只需要设置控制器 Controller 和可选的自定义 GUIEditor。

### 4.1.2 系统 VST 架构设计

VST 架构的 Processer、Controller、GUIEditor 的组成模式，类似于 MVC 模式，Processor 为 Model，负责系统的处理运算，是整个系统中的逻辑实现部分。并且 Processor 的实现并不依赖于 Controller 与 GUIEditor，所以同一个 Processor 可以对应多个 GUIEditor，而 GUIEditor 通过 Processor 返回的数据进行更新显示。

VST 架构中的 Controller 就是控制器，它接受用户输入 GUIEditor 的控制参数，并用过 Processor 处理用户操作、改变 GUIEditor 的视图显示。Controller 的作用是控制程序的操作流程，控制 Processor 的处理作用，再用 GUIEditor 现实返回的信息。

GUIEditor 是 View 视图部件，负责用户与系统的界面交互，负责显示界面上的功能按键和各类元素，所以 GUIEditor 可以有多个，这使 VST 的界面变得多种多样。操作功能也变得丰富。

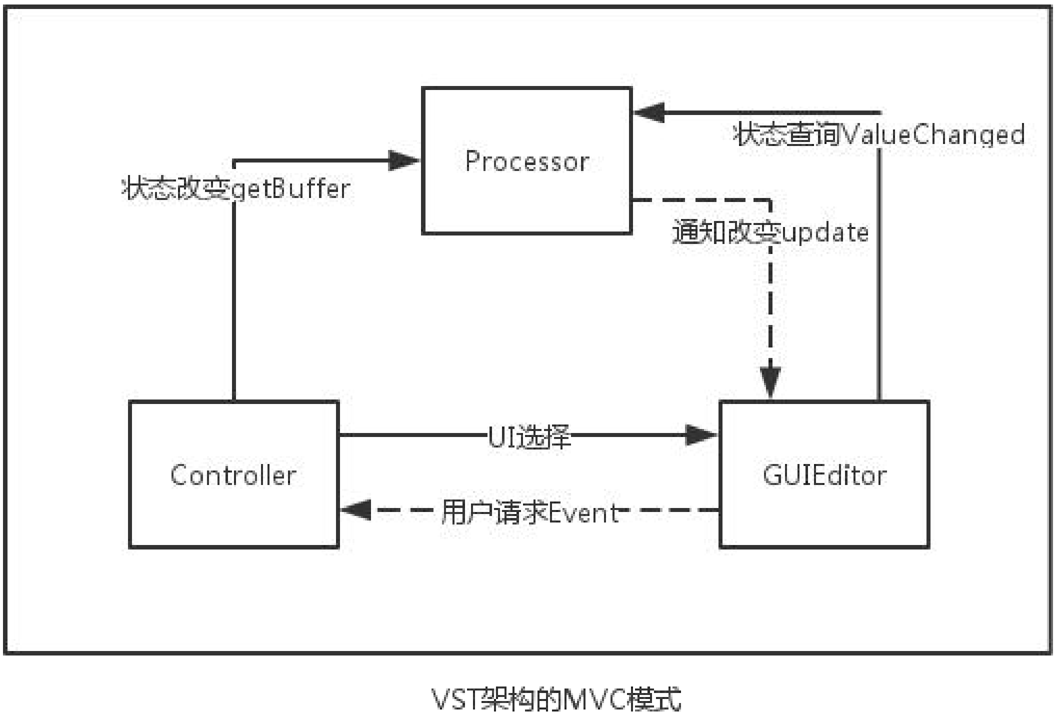


图 4-5 系统 MVC 模式图

本系统通过 GUIEditor 实现系统的功能界面，包括三个模块的 UI 界面和功能视图设计。准确的说，前两个功能模块属于同一个视图，UI 方面无变化。Processor 实现系统的主要功能逻辑部分，包括对音频的效果处理、元数据的编辑储存操作、音频数据的读取、输出操作等。Controller 实现控制 Processor 做相映的处理操作以及控制 GUIEditor 的视图组件进行现实。

## 4.2 系统流程

系统的整体流程是系统完成整体事件需求的驱动，该系统的本质功能为编辑 VR 电影所需的音频内容，从三大功能模块进行分析，可以得到完整的工作流程。

1）用户开始使用该系统，使用 DAW 系统进行，创建音轨并加载该系统；

2）判断 channel 上的音频数据是否存在，用户可为音频做出相应修改，如音频效果添加操作，也可转换音频轨道格式；

3）用户可为音频添加元数据，判断音频是否存在，存在则创建元数据文件；

4）用户编辑元数据，存入数据库并保存元数据文件；

5）用户选择展现音源，选择音频列表中的音频文件、以及元数据文件；

6）Processor 通过 TwirlingAudio 引擎解析元数据并输出环绕声，进行音源展现，用户佩戴耳机进入预览。

# 5 数据库设计

数据库设计的主要目的是建立数据库系统，是必不可少的技术。由 VR 音频内容编辑系统主要包含三大功能模块，该数据库设计与实现主要为了有效存储音频音源还有元数据，作者在实现数据库的设计方面主要考虑音频以及可编辑内容。在接下来的篇幅中，作者主要对数据库进行数据模型的设计，以及相关类的介绍。

## 5.1 数据模型设计

对数据库模型的设计主要从概念模型、逻辑模型、物理模型的步骤进行讨论，VR 音频内容编辑系统主要包括三大主要功能模块，音频效果编辑模块、元数据编辑模块、音源展现模块。其中涉及音轨、音频、元数据、音源对象、场景几个主要实体对象。这里的是展现的是数据库的 E-R 关系图。

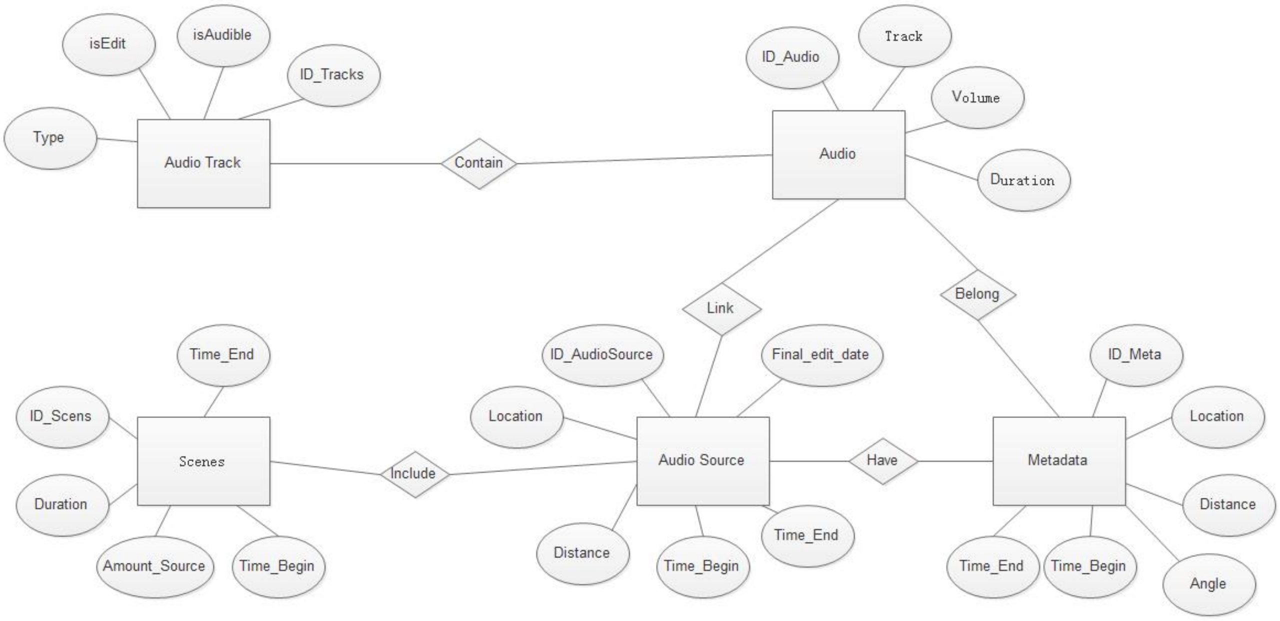


图 5-1 数据库 E-R 图

### 5.1.1 概念模型

建立此数据库的数据概念模型，基本上就是从数据的角度出发，分析系统中数据的架构，包括的得到、传输、处理、储存、输出，通过分析和总结，最后建立起逻辑模型的过程。概念模型的主要功能是描述数据状态，其中包括数据库中的各大主要实体对象。另一方面，概念模型不关心数据库的主要实现方式，而是将重点放在分析数据库各类数据的状态。

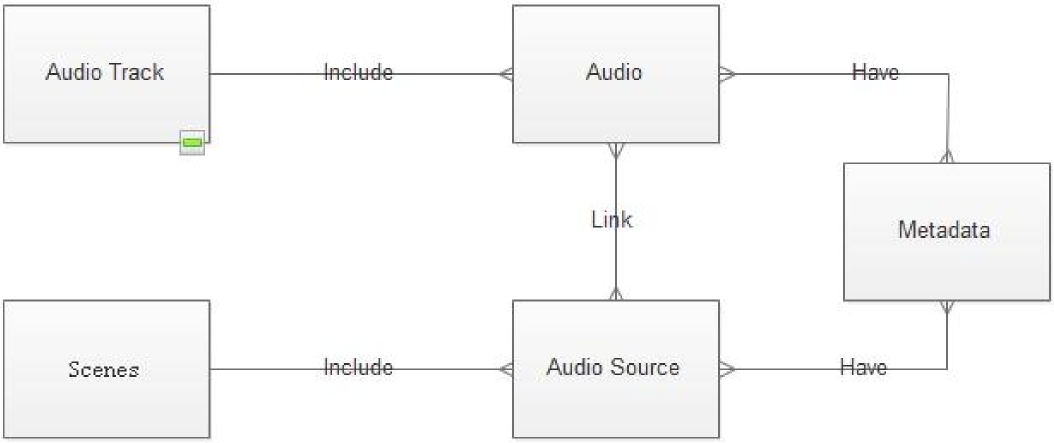


图 5-2 概念模型

可以从概念模型中看到，从系统逻辑分析得出，其中几个主要实体对象。Audio Track 是主要用于实现音频轨道保存音频内容的实体对象，Audio 是主要的音频内容实体对象，包含元数据条目，AudioSource 是在空间中显示音源位置的实体对象，包含该音频的元数据条目。Metadata 是音频和音源的元数据实体对象。

实体关系解释：

1）一个 Audio Track（Channel）可以关联多个 Audio，而一个 Audio 只能关联在一个音轨。所以 Audio Track 和 Audio 是一对多的关系。

2） 一个 Audio 可以 link 入多个 AudioSource 中，而一个音源 AudioSource 也可以包含多个 Audio，所以 Audio 和 Audio Source 是多对多的关系。一个 Audio 可以包含多个 Metadata 条目，而一个 Metadata 条目也可以包含在多个 Audio 中，所以 Audio 和 Metadata 是多对多的关系。

3）一个 AudioSource 可以包含一个元数据，而一个元数据也只可以被一个 AudioSource 包含，所以 AudioSource 与 Metadata 是一对一的关系。

4）一个 Scene 可以包含多个 AudioSource 对象，而一个 AudioSource 对象只可以包含在一个场景中，所以 Scene 与 AudioSource 是一对多的关系。

概念模型的建立对后续模型的建立打下了基础，进一步分析可以得出逻辑模型。

### 5.1.2 逻辑模型

逻辑模型扩展了概念模型，它可以来描述对象系统，逻辑模型主要描述该系统

的主要逻辑，包括该体统具体的工作内容，该系统的主要功能等等。逻辑模型中包括数据库的实体关系和主要属性。

据之前的概念模型设计，可以延伸出数据库的逻辑模型设计。如图 5-3 所示。

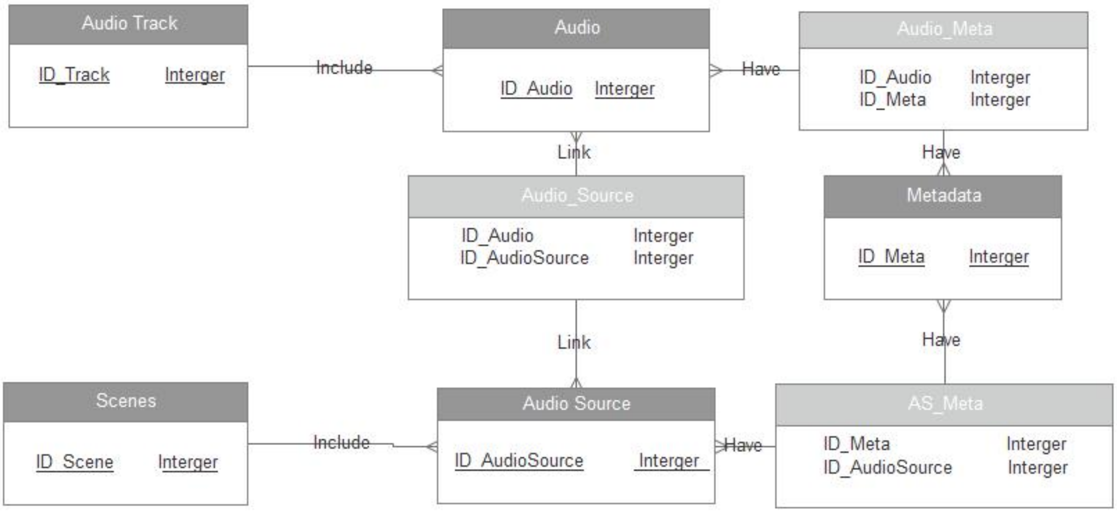


图 5-3 逻辑模型

此逻辑模型中，通过对实体对象的分析，可以加入相关的关系映射形成新的实体。

1）Audio 与 AudioSource 的多对多的关系形成了 Audio_Source，相关实体的主键是新实体的联合主键，Audio_Source 实体包含这两个实体的关系。

2）Audio 与 Metadata 的多对多的关系形成了 Audio_Meta，相关实体的主键是新实体的联合主键，Audio_Meta 实体包含这两个实体的关系。

3）AudioSource 与 Metadata 的多对多的关系形成了 AS_Meta，相关实体的主键是新实体的联合主键，AS_Meta 实体包含这两个实体的关系。

### 5.1.3 物理模型

物理模型又是逻辑模型的扩展和进一步详细，物理模型将考虑到技术实现因素，进行数据库架构设计的过程中，物理模型将真正实现数据在数据库的整体架构。

物理模型包含之前所涉及到的所有实体的所有属性，物理模型的主要目的就是为了实现逻辑模型的数据架构，真正的实现数据库保存数据的功能。

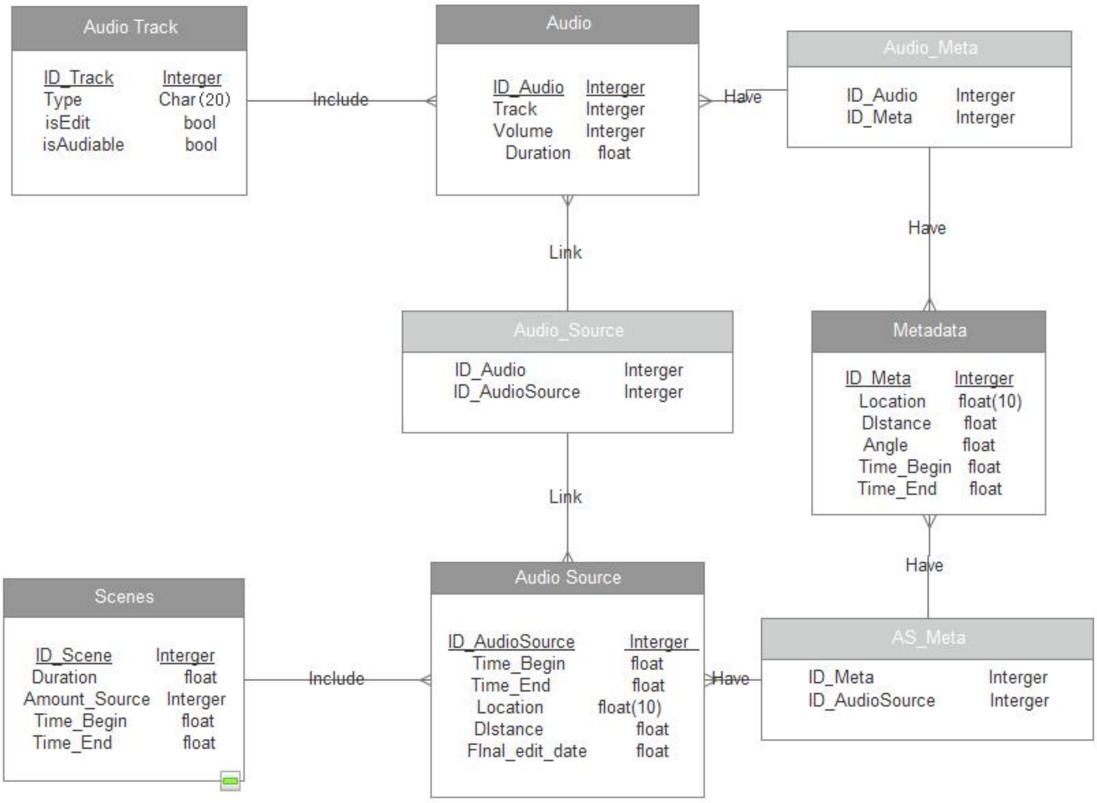


图 5-4 物理模型

## 5.2 本章小结

本章主要介绍数据库的数据模型设计，描述了三个主要模型：概念模型、逻辑模型、物理模型。本章并非该系统的主要实现点，其主要作为音频可编辑数据的管理模块设计出现。

# 6 VR 音频内容编辑系统的详细设计

本系统主要有音频效果编辑模块、Metadata 元数据编辑模块、音源展现模块三个功能模块，本章将从各个模块的功能划分、模块的核心流程、模块的主要类设计三点来阐述该系统的详细设计。

## 6.1 音频效果编辑模块详细设计

音频效果编辑模块是该系统中负责编辑音频内容的部分，主要目标是为实现音频数据编辑功能。该功能主要依靠 VST 架构实现，通过实现 VST 架构的 Processor 与 Controller 模块，实现 GUIEditor 的编辑界面，再通过 Processor 读取音频流实现音频的效果修改后输出的过程。

该模块通过 Processor 读取音频流，用户通过 GUIEditor 与 Controller 进行交互，传入用户的操作参数 parameter，GUIEditor 的参数 parameter 传递给 Processor 对输入的音频数据进行修改，最后输出音频。

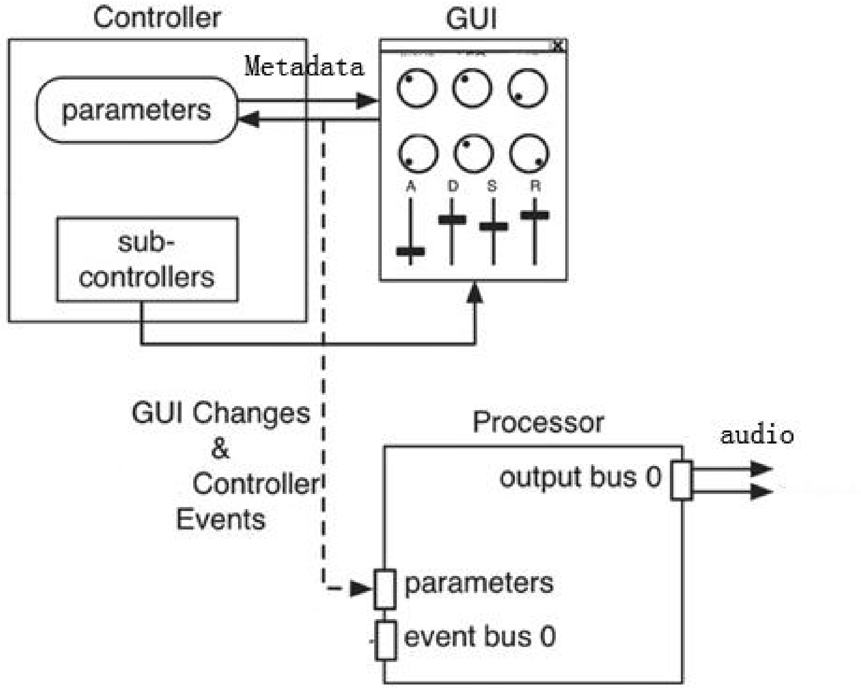


图 6-1 音频效果编辑模块 VST 架构图

### 6.1.1 模块的功能划分

本系统对音频内容的操作主要为音频提供增益、延迟和声道转换的功能。用户使用音频效果编辑模块的过程中，操控 VST 架构的 GUIEditor 进行操作，该模块分为以下几部分：

1）音频增益单元；

用户为音频添加增益 Gain，使音频数据发生增益效果，主要表现为音量改变。

2）音频延迟单元；

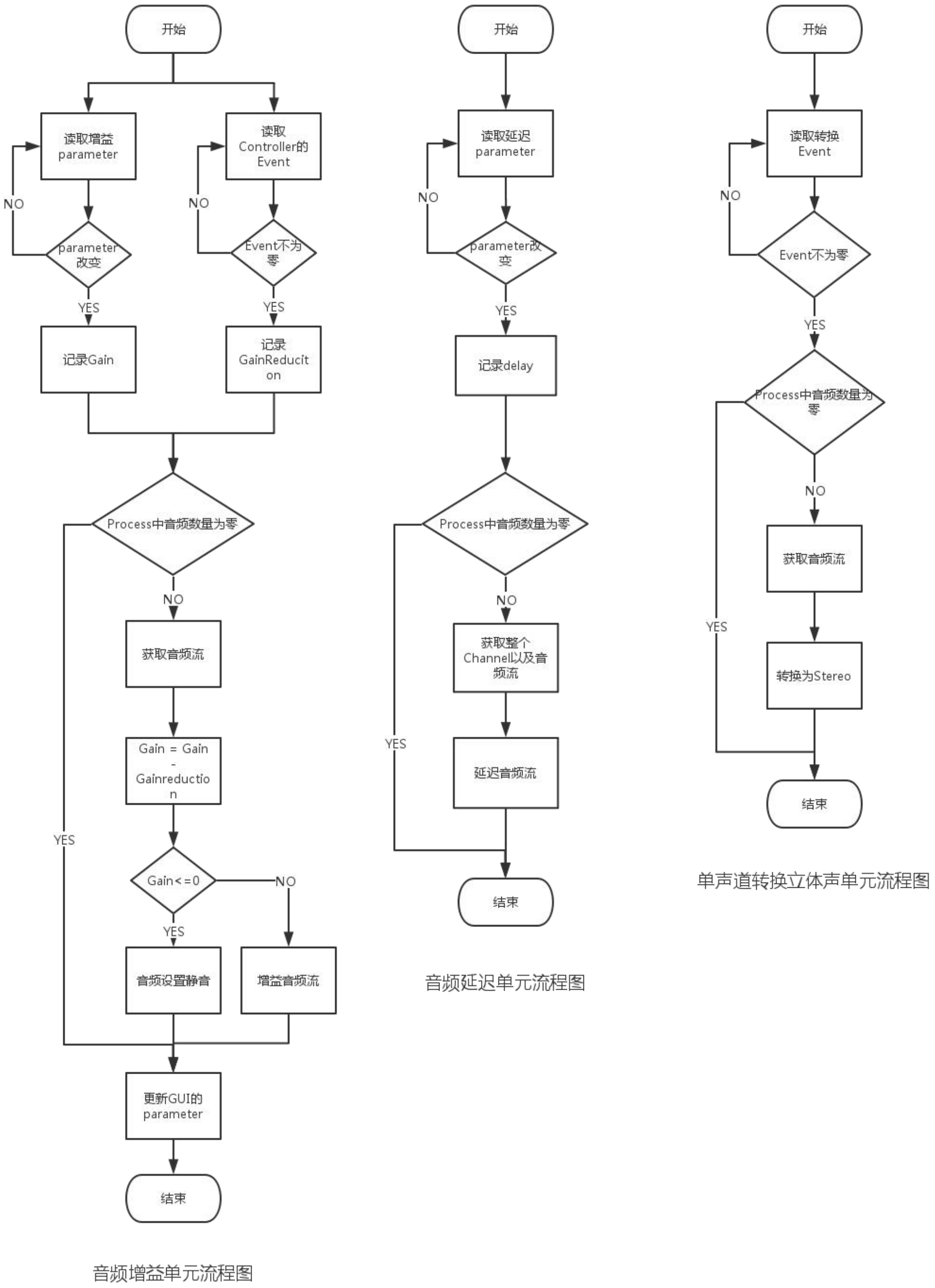
用户为音频添加延迟 Delay，使音频数据发生延迟效果，主要表现为音频播放速度变慢，音频波形数据被拉长，使音频产生延迟的效果。

3）单声道转换立体声单元；

用户转换音频轨道为立体声 Stereo，使单声道音轨变为双声道音轨。

### 6.1.2 模块功能的核心流程

用户在宿主程序 Channel 载入该模块后，用户进入系统界面，界面包含该 Channel 的音频信息，包括音频增益 Gain 滑动条、音频延迟 Delay 滑动条、立体声输出转换按钮、Metadata 元数据添加按钮、Metadata 元数据编辑界面展开按钮，右侧显示 Gain、Delay、声道的参数信息。



音频增益单元流程图

音频延迟单元流程图

单声道转换立体声单元流程图

图 6-2 音频效果编辑模块功能流程图

1）增益音频

用户可以在确保 Channel 中有加载音频资源的情况下利用 VST 架构的效果增

益实时改变音频增益效果，滑动滑动条改变 Gain 的值会为音频流添加增益效果，增加 channelBuffer 的输出值。假如将增益 Gain 的值调到最小，则会静音该 channel，确保增益不出现负数。增益的效果使音频输出音量增加。

2）延迟音频

用户通过滑动延迟 Delay 滑动条为 channel 添加延迟效果，Processor 中读取 channel 的数据流并为其添加延迟参数，使音频播放变缓，音频波形文件的 position 增加，波形数据被拉长。

3）单声道转为立体声输出

用户点击立体声输出转换按钮，会为 channel 转换输出的音频声道属性，转为双声道输出，但是假如为单声道音频文件，则不会产生双声道立体声效果。

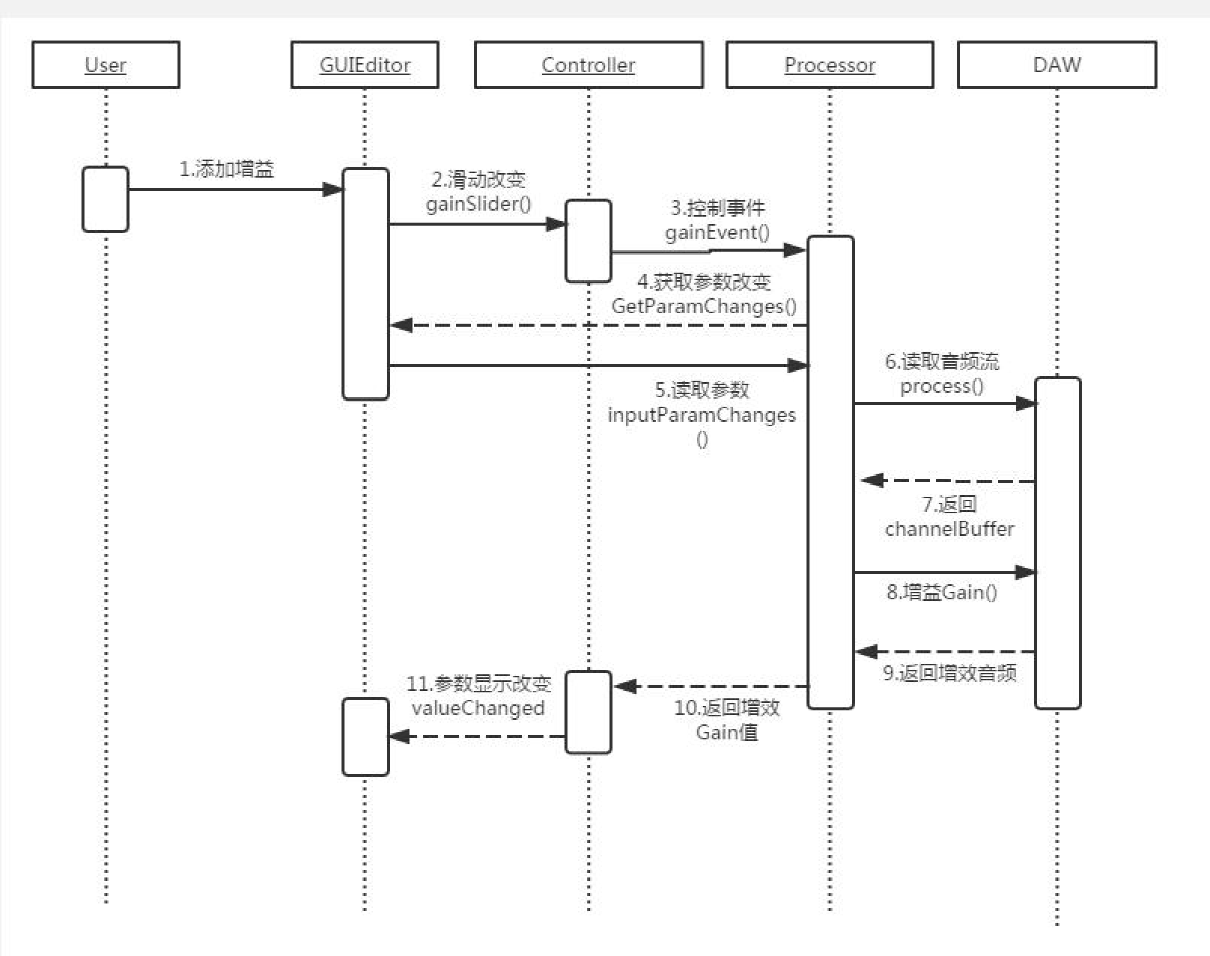


图 6-3 增益音频时序图

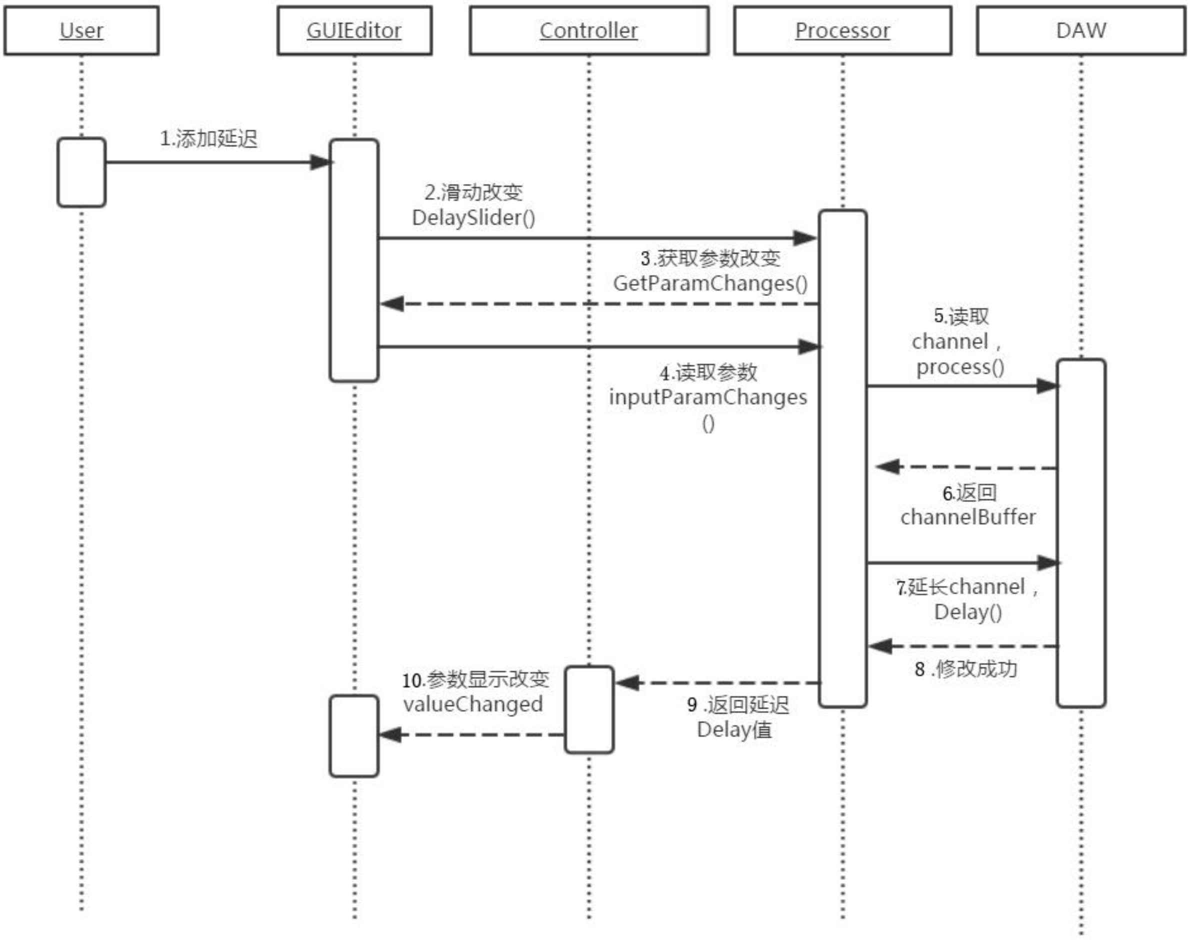


图 6-4 添加延迟效果时序图

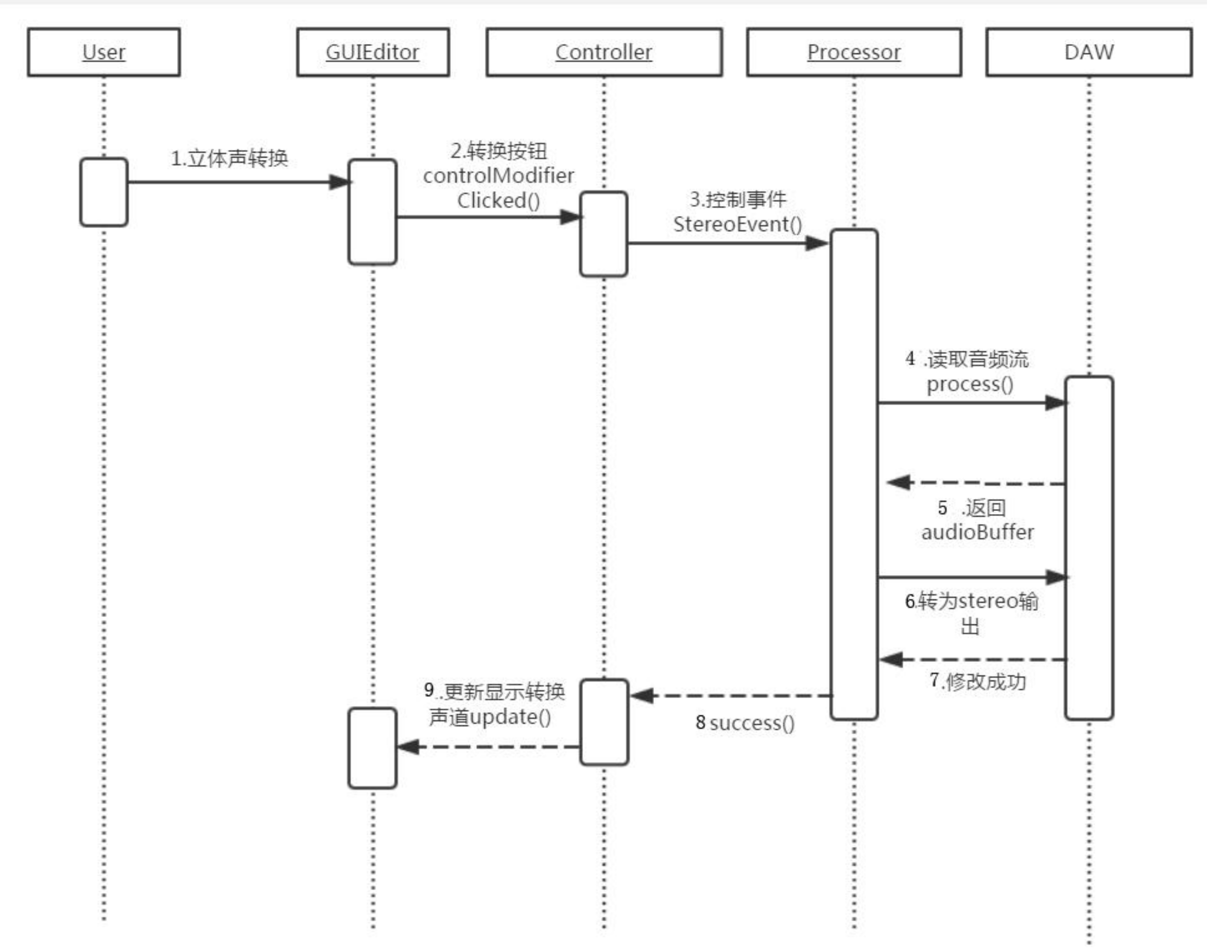


图 6-5 立体声转换时序图

### 6.1.3 模块的类设计

该模块主要使用 VST 框架解决音频效果与基本音频效果编辑功能，VSTGUIEditor 实现系统界面的相关组件，包括滑动条、按钮和 Metadata 编辑界面。AController 实现 VST 架构的 Controller 控制器，Controller 控制器接收 GUI 的控制参数，传入 Processor 进行音频处理，Processor 由 AProcessor 类实现，对整个模块进行初始化，对用户操作进行监听，之后对音频流进行修改。

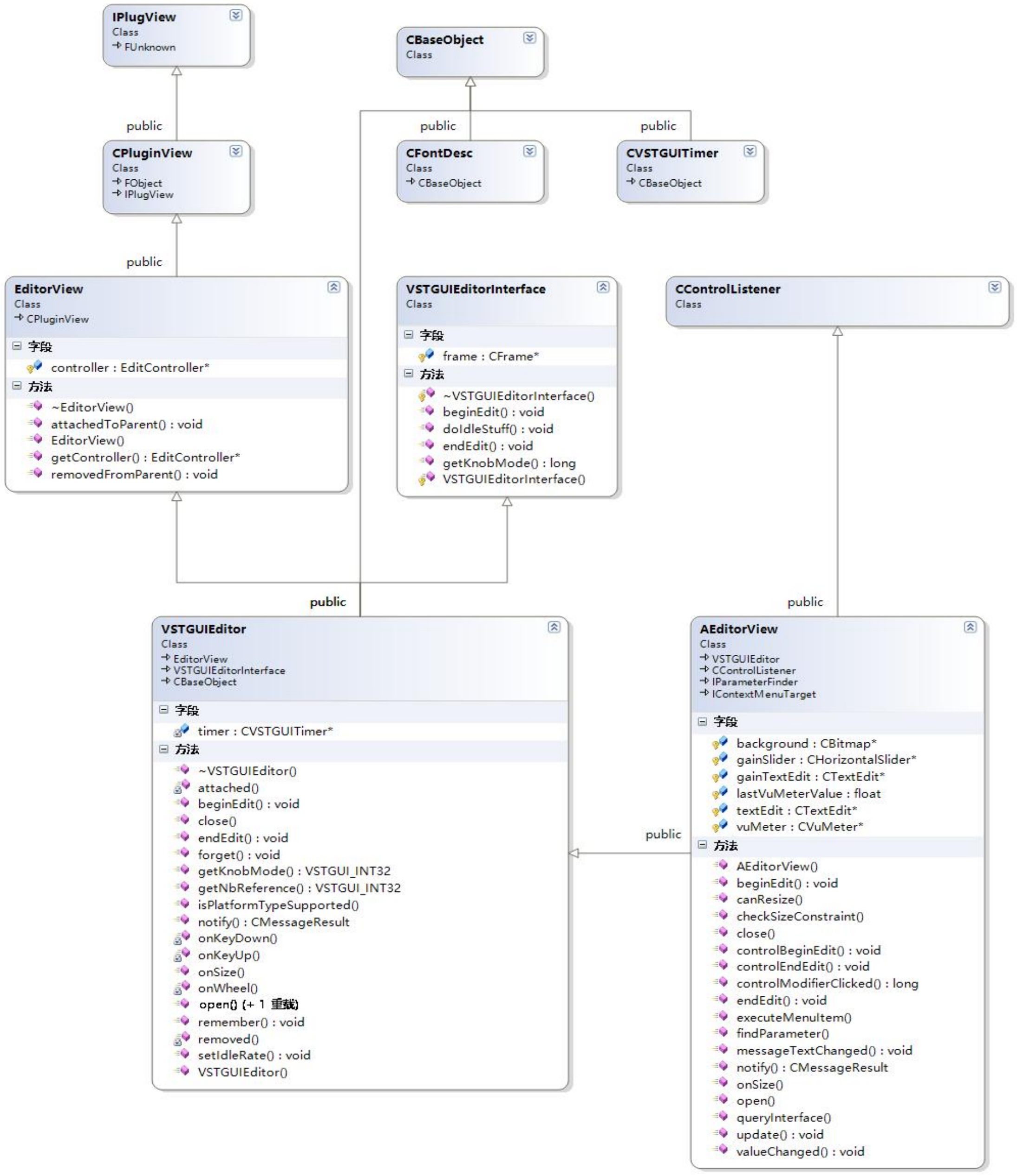


图 6-6 GUIEditor 相关类图

AEditorView 通过对 GUIEditor 的继承，实现 GUI 的基本原件，AEditorView 通过 controlBeginEdit（）和 controlEndEdit（）记录用户每一次控制 GUI 原件的参数变化，并将控制信息传递给 ControlLisener。Processor 通过对 update （） 的调用实现更新 GUI 的操作，messageTextChanged()控制 GUI 上文字编辑原件的数据改变，valueChanged（）控制 GUI 上参数的改变。

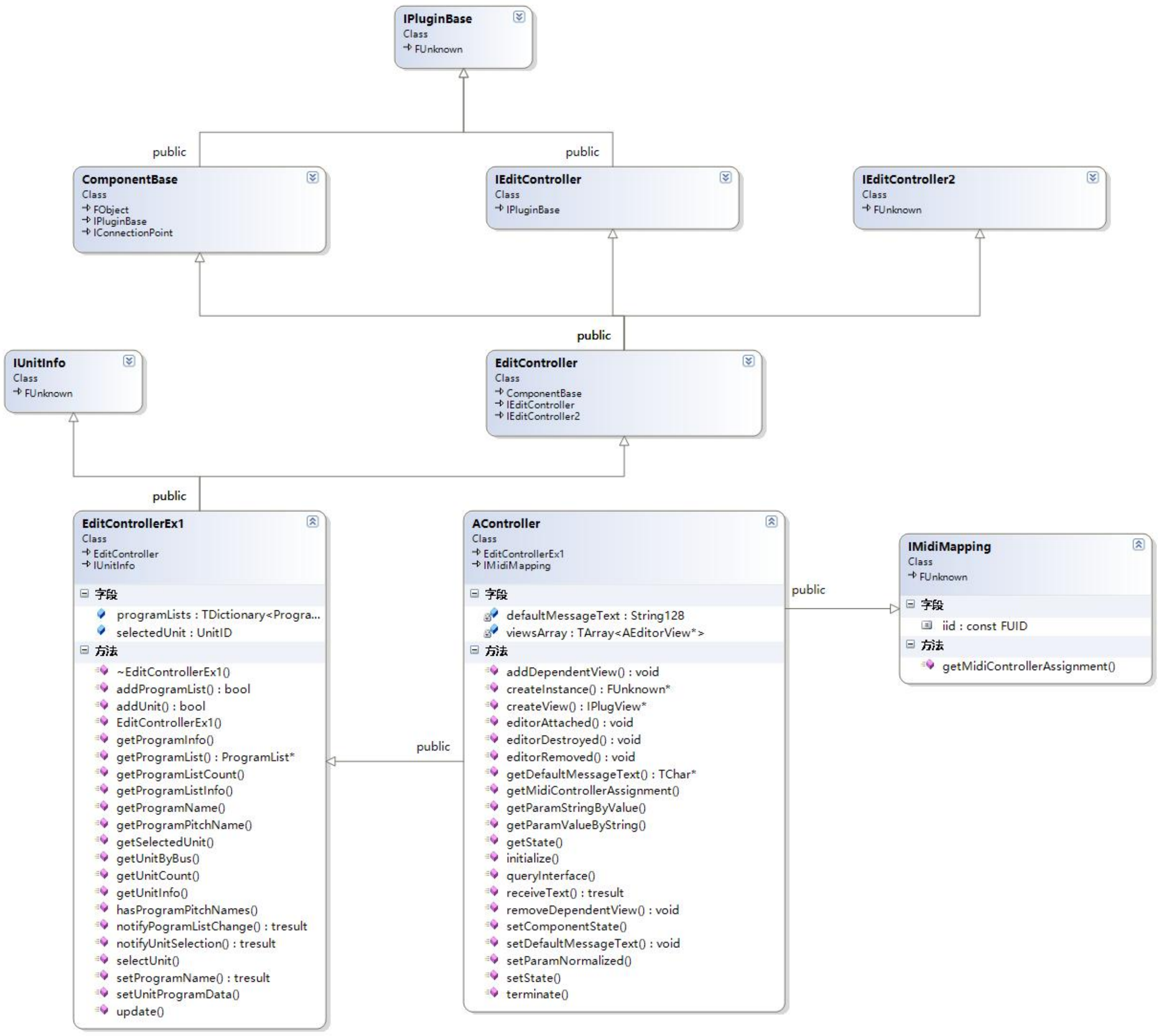


图 6-7 Controller 相关类图

AController 通过对 EditController 的继承，创建 GUI 的窗口和控制组件，实现了控制器的对 GUI 的控制功能，通过控制器接收到 GUI 的文本信息改变和其他参数信息的改变。AController 需要接收 GUI 模块用户输入的 Gain 参数和 Delay 参数，以及修改音频声道信息的操作参数，将它传递给 Processor。

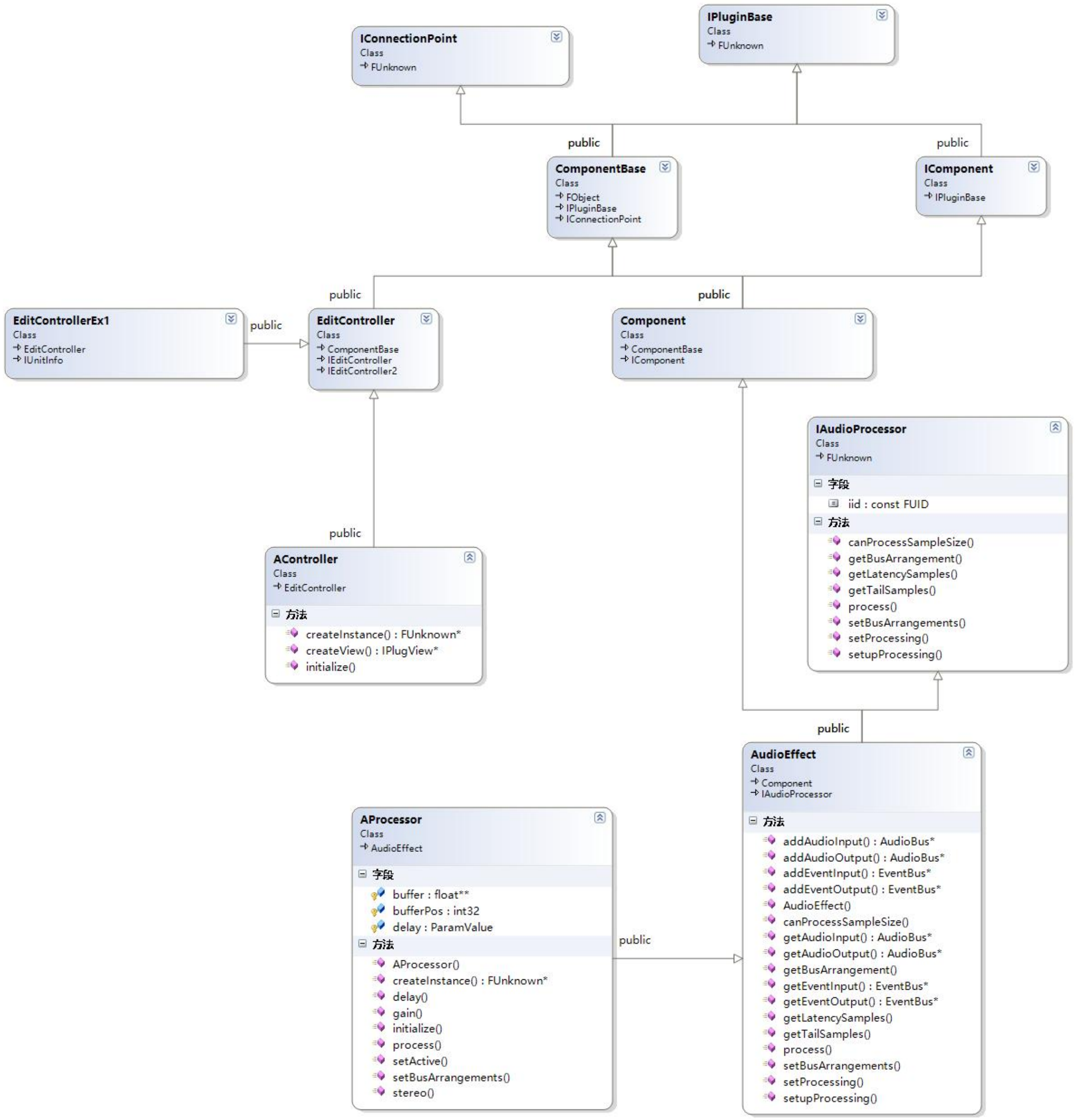


图 6-8 Processor 相关类图

AProcessor 通过 AudioEffect 的 getAudioinput（）读取音频流输入数据，通过 getAudioOutput 读取输出的音频流，AProcessor 通过 gain（）与 delay（）、stereo（）对音频流进行修改，通过 process（）对音频流进行输出。当增益参数极小，音频流将静音，增益参数不能为负值，该部分在 gain（）中实现。添加部分代码。

```
if (gain < 0.0000001)

{

int32 sampleFrames = data.numSamples;

for (int32 i = 0; i < numChannels; i++)
{
memset (out[i], 0, sampleFrames * sizeof (float));
}
data.outputs[0].silenceFlags = (1 << numChannels) - 1;
}
else
{
for (int32 i = 0; i < numChannels; i++)
{
int32 sampleFrames = data.numSamples;
float* ptrIn  = in[i];
float* ptrOut = out[i];
float tmp;
while (--sampleFrames >= 0)
{
    // apply gain
    tmp = (*ptrIn++) * gain;
    (*ptrOut++) = tmp;
    // check only positive values
    if (tmp > fVuPPM)
    {
    fVuPPM = tmp;
    }
}
}
}
```

## 6.2 Metadata 元数据编辑模块详细设计

在公司内部，元数据解决了VR音频与VR视频的对接问题。在该模块中，用户可通过VST实现的元数据Metadata编辑功能来对该音频的Metadata进行编辑，通过其中几个属性的修改，编辑后点击入库按钮保存该音频与其元数据。元数据与已添加的音频文

件关联，同时也与该音轨关联，用户可对音轨元数据进行添加和编辑，这个元数据可用来定位音源在空间中的位置，如此一来则使后期编辑工作效率更加快速。元数据信息包括音频时长、音源空间坐标、开始时间、结束时间、开始横纵向角度。

### 6.2.1 模块的功能划分

元数据添加允许用户给音轨添加元数据文件，编辑元数据文件后关联元数据与音轨对象，音轨中的音频文件同时关联元数据对象。音轨的元数据编辑需要用户对音源在空间中的相对位置进行设置，这个过程一般在用户看到VR视频与音频形成对照之后才可以进行，所以元数据的编辑一般需要元数据储存，避免元数据被以任何途径的修改，导致的数据丢失。

该模块包含元数据添加，元数据编辑、元数据存储、元数据列表显示四点功能。

1）元数据添加

选择音轨加载该模块后，添加元数据为音频文件添加元数据文件。

2）元数据编辑

用户可编辑该添加的元数据，元数据编辑中包含元数据ID、元数据音频ID、音源空间坐标、音源距离、音源起始角度、音频开始时间点、音频结束时间点、音频时长。

3）元数据存储

用户编辑元数据完成后，点击入库SaveData按钮保存元数据，保存后元数据将不可继续修改，但可以为音频添加新的元数据。

4）元数据列表显示

元数据编辑模块右侧显示元数据列表，列表内容显示当前存在库中的元数据。显示元数据的ID和音源ID。

### 6.2.2 模块的功能核心流程

元数据编辑模块主要流程比较简单，添加元数据后便可修改元数据设置参数，用户在编辑好元数据后就可点击SaveData按钮进行储存。

1）添加元数据

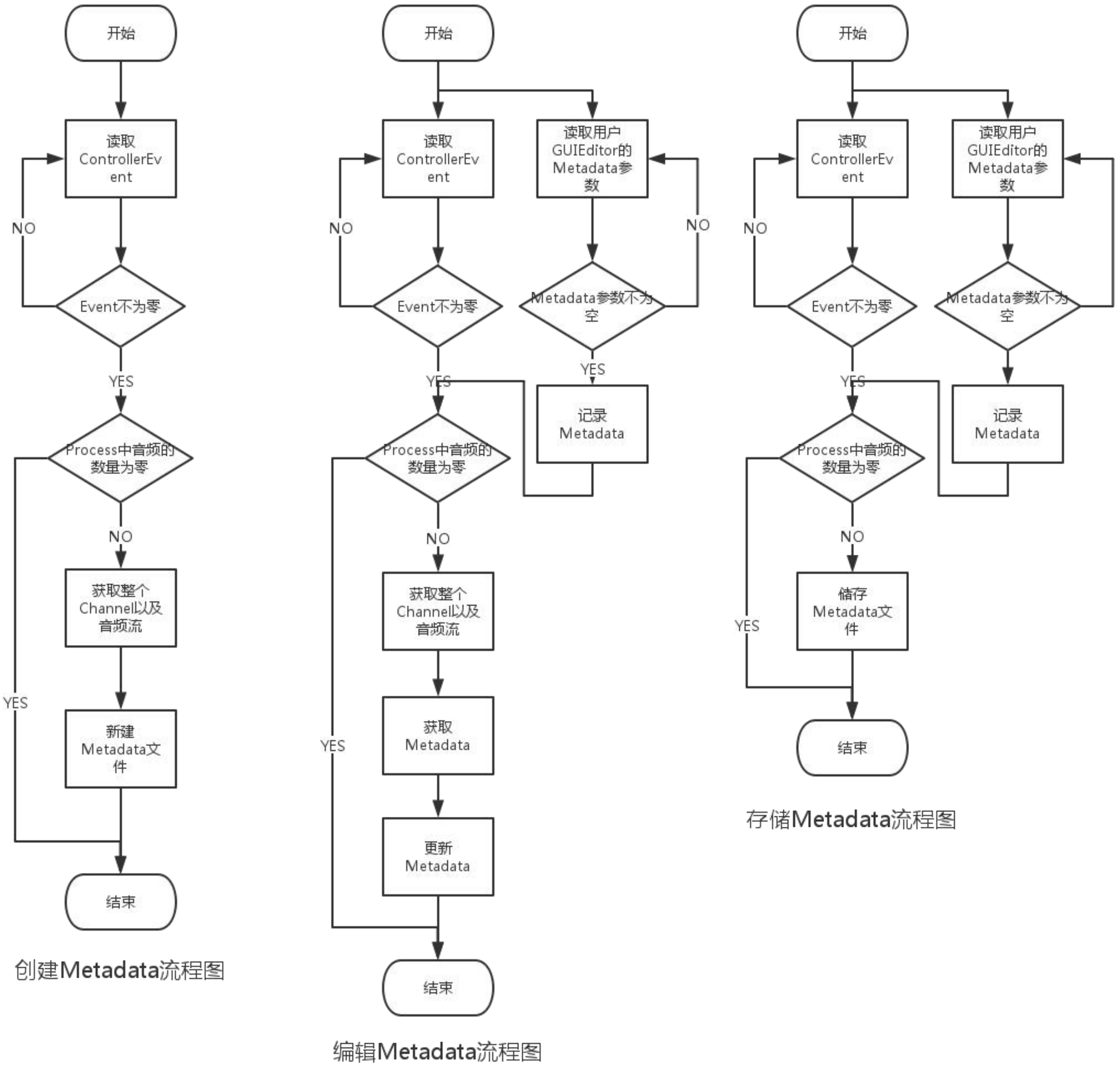
用户进入系统后，点击下方的展开按钮，展开元数据编辑模块。用户先点击添加元数据，保证音轨有音频，之后元数据编辑列表会变为可编辑状态，没有元数据的情况下为不可编辑状态。添加元数据会为音频新建元数据文件metadata.txt。因为公司视频SDK的专有元数据解析模块只解析txt文件，所以保存txt格式的元数据文件。添加元数据，Processor读取Controller的Event，假如此时音频数据为空，则无法添加元数据。假如不为空，则获取整个Channel，并新建Metadata文件，在数据库中添加元数据文件。返回元数据列表，通过GUIEditor显示在界面列表。

2）编辑元数据

元数据编辑列表中包含元数据 ID、元数据音频 ID、音源空间坐标、音源距离、音源起始角度、音频开始时间点、音频结束时间点、音频时长。用户根据 VR 电影的要求编辑元数据。Processor 读取 GUIEditor 的元数据参数，通过 Controller 获取元数据的显示，更新元数据信息。

3）保存元数据

用户编辑元数据完成后，点击入库 SaveData 按钮保存元数据，保存后元数据将不可继续修改，可以为音频添加新的元数据。



创建Metadata流程图

编辑Metadata流程图

存储Metadata流程图

图 6-9 元数据编辑模块功能流程图

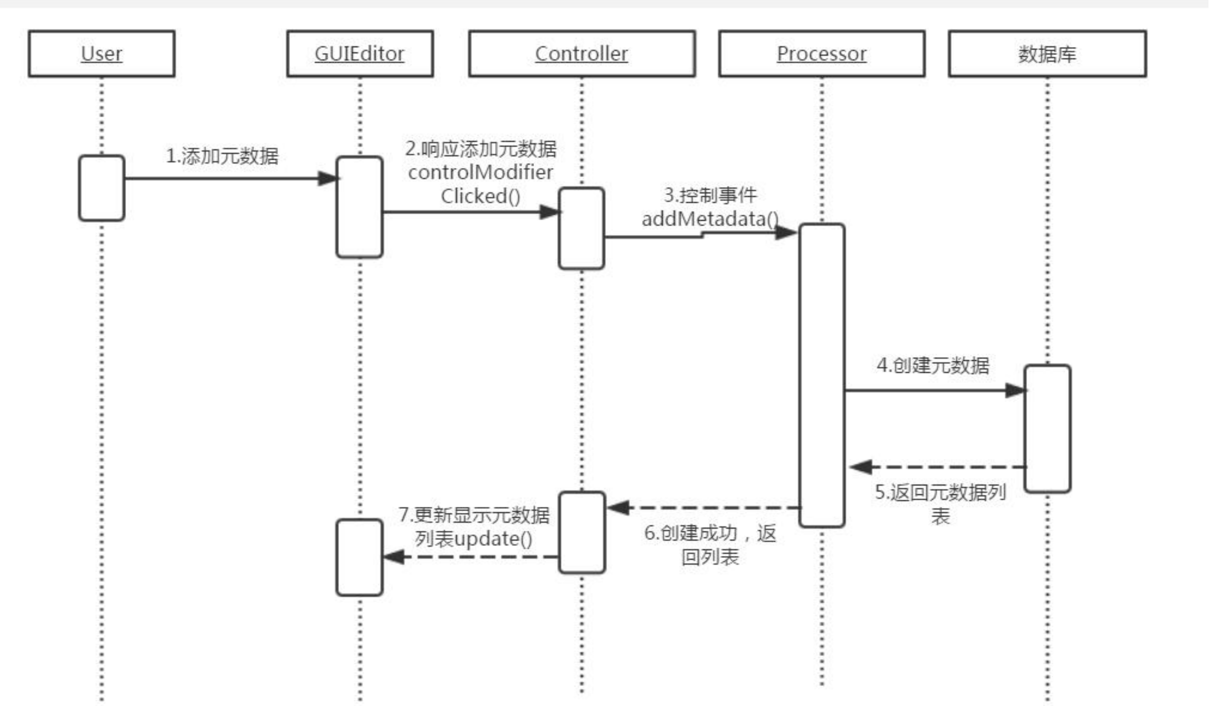


图 6-10 添加元数据时序图

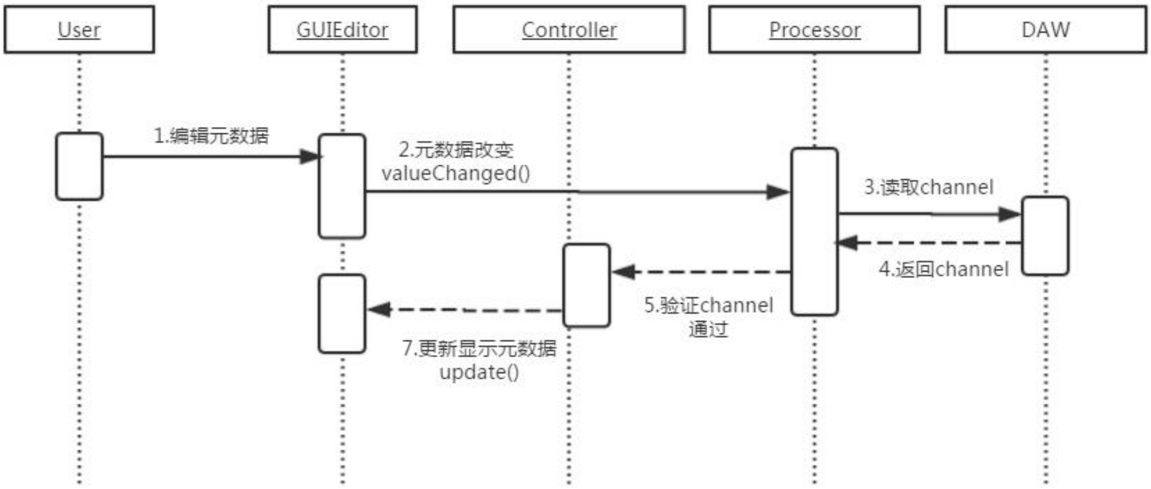


图 6-11 编辑元数据时序图

### 6.2.3 模块的类设计

AMetadata 类继承 AMetaEditor，AMetadata 类中包含元数据的数据类型，实现 AMetaEditor 的 process（）方法，读取 channel 判断，channel 不为空则添加元数据，为空则返回无效。编辑元数据时，process（）对 GUIEditor 的参数进行读取，metadata 参

数改变时，记录元数据。

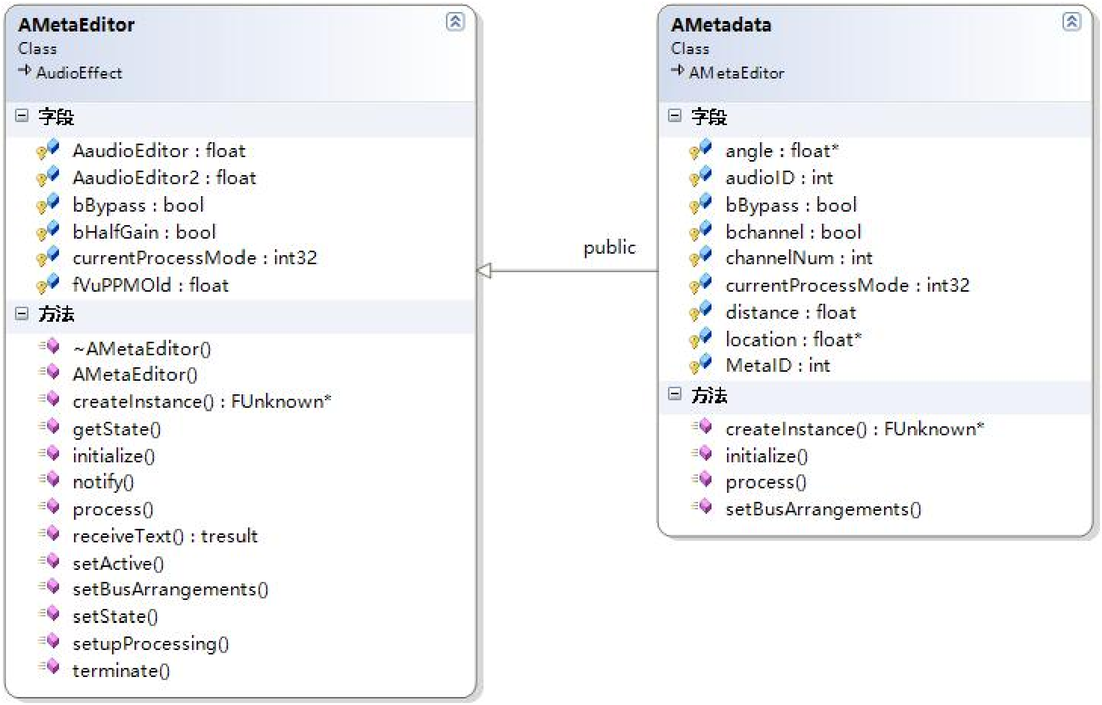


图 6-12 元数据相关类图

## 6.3 音源展现模块详细设计

进行音频的元数据测试工作时，通过此模块实现音源音频播放展现。本模块结合公司的全景声 TwirlingAudio 引擎，通过该引擎读取音频将环绕声展现出来。通过之前设置的 metadata 和编辑的音频文件通过此引擎解析 Metadata 元数据并定位虚拟音源。

### 6.3.1 模块的功能划分

音源展现模块包含三个主要功能：

1）音频和元数据列表显示

用户进入系统后，点击预览按钮，出现音频列表和元数据列表。用户可通过音频和元数据列表选择进行音源展现的音频和元数据。选择音频、元数据，点击展现按钮，提示展现成功，则可以进行音源的播放。

2）音源播放

展现成功后，点击播放按钮，提示需要带耳机设备进行音源展现，戴上耳机后可以听到虚拟音源的环绕声循环播放，模拟 VR 电影中的音源位置。

3）元数据调整

在元数据编辑栏里调整元数据，调整完成后再次点击展现，展现成功后再次播放可以测试刚才的元数据是否调整正确，点击 SaveData 按钮储存调整后的 Metadata。

### 6.3.2 模块的功能核心流程

1）音频和元数据列表显示

用户进入系统后，此时已确定 Process 中有音频文件或者库中有音频文件，点击预览按钮，响应 ControllerEvent，Processor 从数据库中读取音频和元数据表。更新 GUIEditor 显示音频列表和元数据列表。用户选择列表中的音频和元数据，Processor 通过 valueChanged（）获取 GUIEditor 的选择参数变化。得到选择音频、元数据，点击展现按钮，通过 TwirlingAudio 读取音频和元数据 metadata，提示展现成功，点击播放进行音源的播放。

2）音源播放

在 TwirlingAudio 中，读取音频流，获得音频轨道、大小、码率，TwirlingAudio 特殊要求音频码率只支持 44100bps，判断是否为 44100 码率。假如符合，则对音频按照读入的 Metadata 进行环绕音输出。

3）元数据调整

在元数据编辑列表中编辑元数据，Processor 通过 valueChanged（）获取新的元数据参数，用户点击 SaveData，响应 ControllerEvent 储存 Metadata 文件，更新数据库。

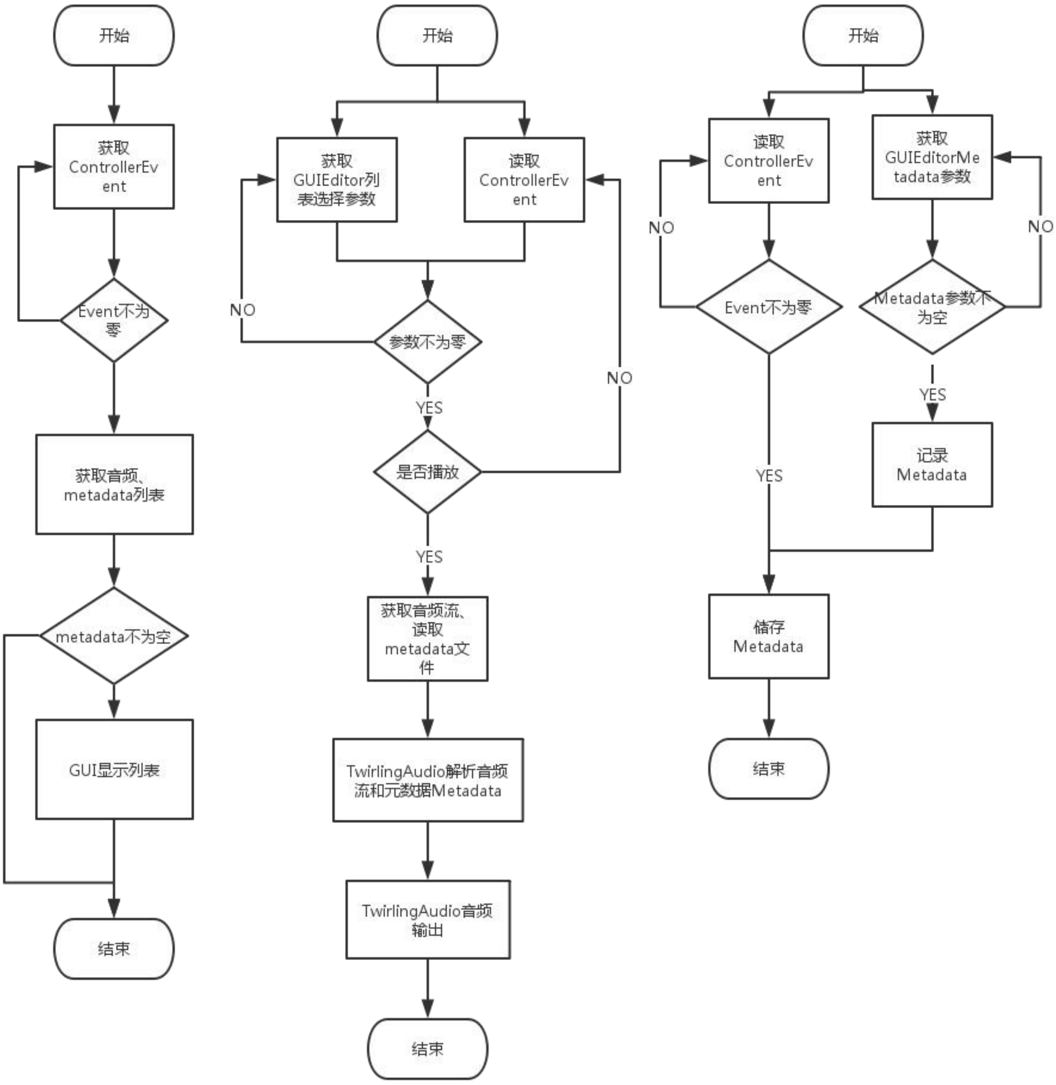
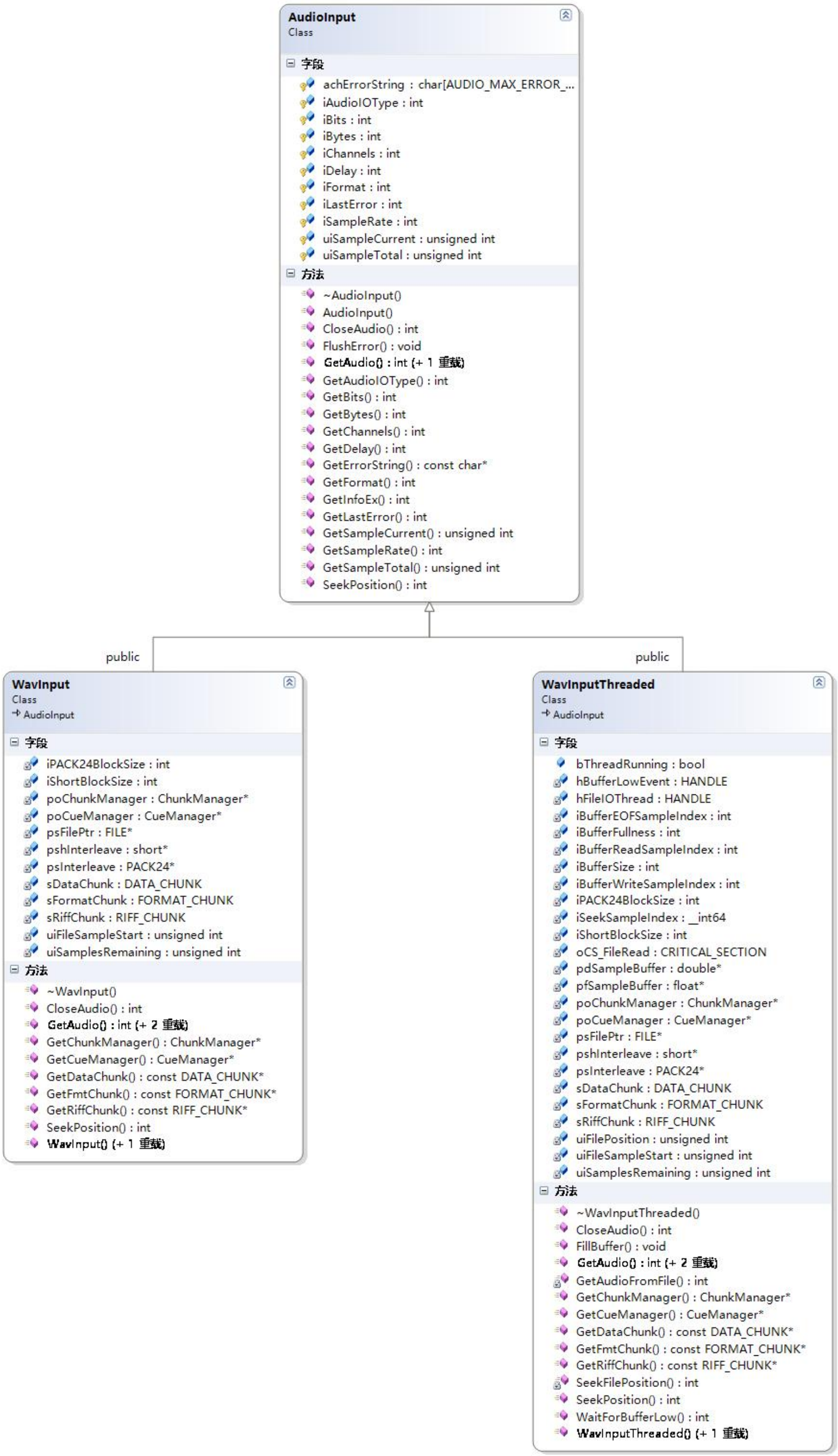


图 6-13 音源展现模块主要功能流程图

### 6.3.3 模块的类设计


AudioInput
Class
字段
achErrorString : char[AUDIO_MAX_ERROR_...
iAudioIOType : int
iBits : int
iBytes : int
iChannels : int
iDelay : int
iFormat : int
iLastError : int
iSampleRate : int
uiSampleCurrent : unsigned int
uiSampleTotal : unsigned int
方法
~AudioInput()
AudioInput()
CloseAudio() : int
FlushError() : void
GetAudio() : int (+ 1 重载)
GetAudioIOType() : int
GetBits() : int
GetBytes() : int
GetChannels() : int
GetDelay() : int
GetErrorString() : const char*
GetFormat() : int
GetInfoEx() : int
GetLastError() : int
GetSampleCurrent() : unsigned int
GetSampleRate() : int
GetSampleTotal() : unsigned int
SeekPosition() : int
public
public
WavInput
Class
AudioInput
字段
iPACK24BlockSize : int
iShortBlockSize : int
poChunkManager : ChunkManager*
poCueManager : CueManager*
psFilePtr : FILE*
pshInterleave : short*
psInterleave : PACK24*
sDataChunk : DATA_CHUNK
sFormatChunk : FORMAT_CHUNK
sRiffChunk : RIFF_CHUNK
uiFileSampleStart : unsigned int
uiSamplesRemaining : unsigned int
方法
~WavInput()
CloseAudio() : int
GetAudio() : int (+ 2 重载)
GetChunkManager() : ChunkManager*
GetCueManager() : CueManager*
GetDataChunk() : const DATA_CHUNK*
GetFmtChunk() : const FORMAT_CHUNK*
GetRiffChunk() : const RIFF_CHUNK*
SeekPosition() : int
WavInput() (+ 1 重载)
WavInputThreaded
Class
AudioInput
字段
bThreadRunning : bool
hBufferLowEvent : HANDLE
hFileIOThread : HANDLE
iBufferEOFSampleIndex : int
iBufferFullness : int
iBufferReadSampleIndex : int
iBufferSize : int
iBufferWriteSampleIndex : int
iPACK24BlockSize : int
iSeekSampleIndex : _int64
iShortBlockSize : int
oCS_FileRead : CRITICAL_SECTION
pdSampleBuffer : double*
pfSampleBuffer : float*
poChunkManager : ChunkManager*
poCueManager : CueManager*
psFilePtr : FILE*
pshInterleave : short*
psInterleave : PACK24*
sDataChunk : DATA_CHUNK
sFormatChunk : FORMAT_CHUNK
sRiffChunk : RIFF_CHUNK
uiFilePosition : unsigned int
uiFileSampleStart : unsigned int
uiSamplesRemaining : unsigned int
方法
~WavInputThreaded()
CloseAudio() : int
FillBuffer() : void
GetAudio() : int (+ 2 重载)
GetAudioFromFile() : int
GetChunkManager() : ChunkManager*
GetCueManager() : CueManager*
GetDataChunk() : const DATA_CHUNK*
GetFmtChunk() : const FORMAT_CHUNK*
GetRiffChunk() : const RIFF_CHUNK*
SeekFilePosition() : int
SeekPosition() : int
WaitForBufferLow() : int
WavInputThreaded() (+ 1 重载)

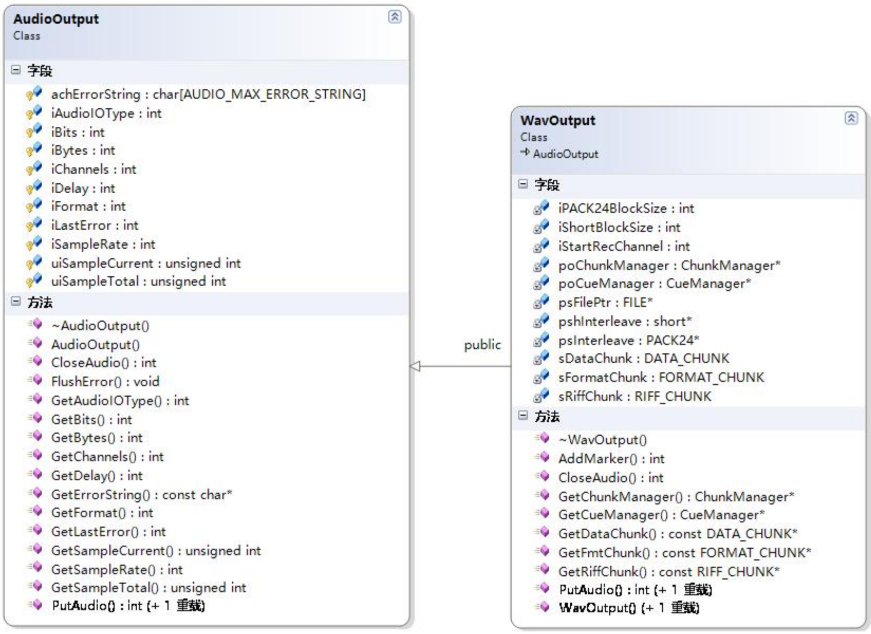


图 6-14 音源展现模块相关类图

## 6.4 本章小结

本章对系统进行了详细设计与实现介绍，包括三个主要功能模块的功能划分、功能核心流程与模块类的设计。对每一个功能模块使用流程图进行功能点的描述以及实现方法的概述，使用时序图对系统功能多个对象之间的协作进行了描述，最后展示完成的各个类图。

# 7 结论

作者通过这次在公司的实习工作，学习掌握到了 VR 电影的制作流程，通过与团队的相互磨合、相互促进与鼓励，完成了项目内所承担的相关工作，对于我个人来说是一次相当有意义的工作经历。同时，通过前期的快速学习，也锻炼了我快速适应工作节奏的能力。前期通过对 VST 架构的学习，笔者慢慢感受到了文献资料的重要性，关于 VST 架构可以参考的资料寥寥可数，这对于项目研究来说无疑是巨大的考验。在团队的帮助下，快速对 VST 架构进行研究，通过几个月的时间，成功对系统有了初步的设计。

该项目应用于公司内部后期工作人员使用，笔者与后期工作人员用一周的时间对项目的主要需求进行分析研究，得出系统的几点基础功能和需求设计草稿。在后面的几个月中，不断的对系统的需求进行分析，与用户交流磨合的过程中不断深入。最终使系统的功能需求完全满足用户的实际需要，达到预期的工作效率。

对于 VR 电影来讲，沉浸感的体验基于视频和音频的结合，准确的结合音视频实现的 VR 电影具有极强的沉浸感。用户常用到的 Cubase 音频编辑工作站兼容 VST 架构的插件，甚至成为了 VST 插件的基本宿主之一。通过 VST 架构开发的音频编辑软件具有符合用户非功能性需求的保证，首先 VST 架构符合用户友好的特征，类效果器的界面展现，完全遵循音乐编辑工作者的日常使用规范；其次，该 VR 音频内容编辑系统可以保证用户的使用效率。

在整个项目的开发过程中，确实存在一些难以解决的问题，对于系统的功能具体实现还缺乏完善，主要体现在功能模块划分存在缺陷，系统的实现平台没有整合，一些可见的漏洞没有解决。这也暴露出笔者在系统设计方面的巨大欠缺，笔者将会在以后的工作中继续努力，加强自己对于系统设计的分析能力。

本项目对于加快公司业务有着良好的推进作用，VR 内容制作的进度加快后将直接推动 VR 电影的制作效率提升。但本系统仅为公司内部使用，希望将来可以进一步进行研究。

# 参考文献

[1] 敖腾河，谢辉，阮宏伟，AO Teng-he，XIE Hui，Ruan Hong-wei . VST/VSTi 构架下网络音源系统的设计与实现 . 内蒙古大学学报(自然科学版) . 2009,40(2).

[2] 王钢， 刘晓莎 . 电影杜比全景声创作初探 . 现代电影技术 . 2014(5).

[3] 林建平， 杨吉慧， Lin Jianping， Yang Jihui . 浅谈 VST 在 MIDI 制作中的应用 . 科技广场 . 2007(9).

[4] 戴云，孙军，王兴东 . 音频编辑处理系统的设计与实现 . 电声技术 . 2004, (10).

[5] Ann Franchesca B. Laguna； Nicanor Marco P. Valdez； Rowena Cristina L. Guevara . MIDI Implementation of a Kulintang Modal Synthesizer using the VST2.4 Standard . TENCON 2012 IEEE Region 10 Conference. [v.2] . Cebu(PH).

[6] 杜比实验室 . 杜比全景声影院扬声器位置设计指南 . 现代电影技术 . 2013(3).

[7] Thompson, J.；Kuchera-Morin, J.；Novak, M.；Overholt, D.；Putnam, L.；Wakefield, G.；Smith, W. The Allobrain: An interactive, stereographic, 3D audio, immersive virtual world . International journal of human-computer studies . 2009, 67(11).

[8] Kai-Uwe Doerr；Rademacher H.；Huesgen S.；Kubbat W. Evaluation of a Low-Cost 3D Sound System for Immersive Virtual Reality Training Systems . IEEE transactions on visualization and computer graphics . 2007, 13(2).

[9] LaViola J.J. Bringing VR and Spatial 3D Interaction to the Masses through Video Games . IEEE Computer Graphics and Applications . 2008, 28(5).

[10] Wang Tao， An Wenguang . The Application Research of Web3D-Based Virtual Reality Technology in Modern Distance Education . 中国江西南昌 . 2011-06-20 .

[11] CHEN Dalei. The Application of Virtual Reality in Art Design: A New Approach[A]. Proceedings of 2015 International Conference on Education Technology,Management and Humanities Science(ETMHS 2015)[C]. 2015.

[12] Kunihiro Nishimura；Aoi Ito；Tomohiro Tanikawa；Michitaka Hirose . VR Based Movie Watching Method by Reproduction of Spatial Sensation . Virtual and Mixed Reality Virtual and Mixed Reality: Third International Conference, VMR 2009 Held as Part of HCI International 2009 San Diego, CA, USA, July 19-24, 2009 Proceedings . San Diego, CA, USA . 2009 年 1 月 1 日.

[13] Doron Friedman；Yishai A. Feldman；Ariel Shamir；Tsvi Dagan . Automated Creation of Movie Summaries in Interactive Virtual Environments . 2004 IEEE Virtual Reality (VR 2004) . Chicago, LLLinois, USA . 2004 年 1 月 1 日.

[14] Takafumi Koike；Kei Utsugi；Michio Oikawa . VR content platform for multi-projection displays with realtime image adjustment . Proceedings of the 2005 international conference on Augmented tele-existence . Christchurch(NZ) . 2005 年 1 月 1 日.

# 致 谢

本片论文是在 XXX 的耐心指导下完成的。初期的论文选题、开题到中期答辩，再到论文的初稿、终稿每一步 XXX 都严谨教学，非常耐心的帮助我解决论文上的写作难题，指导我要写出一片优秀的本科毕业生论文。从论文的写作意义，到论文的写作方式，再到论文的写作规范，每一点 XXX 的指点都为我拨开迷雾，让我对论文写作有了全新的认识。不仅如此，XXX 对学生的谆谆教导以及渊博的知识无疑都让我感到钦佩。在一些学习认识方面，XXX 的深刻理解让我受益匪浅。

在公司的实习工作中，还要感谢公司导师 Andy 对我的全力帮助，以及音频后期 HongXu 不厌其烦的与我讨论项目需求以及其他的帮助，是你们的帮助让我快速融入了团队中去。在此对所有帮助过我的人表示感谢。

此外，还对周围共同奋斗的同学表示感谢，在生活中点滴的帮助，以及相互的鼓励、激励都成为了我向前的动力。

最后感谢评阅毕业设计论文的各位评阅官们。论文中还有不少的浅显见解和不足，我将在今后的学习工作中加倍努力。

# 附　　录

## 附录 A 外文翻译原文

# Virtual Reality System with Integrated Sound Field Simulation and Reproduction

**Tobias Lentz,1 Dirk Schroder,¨ 1 Michael Vorlander,¨ 1 and Ingo Assenmacher2**

*1 Institute of Technical Acoustics, RWTH Aachen University, Neustrasse 50, 52066 Aachen, Germany*

*2 Virtual Reality Group, RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, Germany*

Received 1 May 2006; Revised 2 January 2007; Accepted 3 January 2007

Recommended by Tapio Lokki

A real-time audio rendering system is introduced which combines a full room-specific simulation, dynamic crosstalk cancellation, and multitrack binaural synthesis for virtual acoustical imaging. The system is applicable for any room shape (normal, long, flat, coupled), independent of the a priori assumption of a diffuse sound field. This provides the possibility of simulating indoor or outdoor spatially distributed, freely movable sources and a moving listener in virtual environments. In addition to that, near-tohead sources can be simulated by using measured near-field HRTFs. The reproduction component consists of a headphone-free reproduction by dynamic crosstalk cancellation. The focus of the project is mainly on the integration and interaction of all involved subsystems. It is demonstrated that the system is capable of real-time room simulation and reproduction and, thus, can be used as a reliable platform for further research on VR applications.

## 1. INTRODUCTION

Virtual reality (VR) is an environment generated in the computer with which the user can operate and interact in real time. One characteristic of VR is a three-dimensional and multimodal interface between a computer and a human being. In the fields of science, engineering, and entertainment, these tools are well established in several applications. Visualization in VR is usually the technology of primary interest. Acoustics in VR (auralization, sonification) is not present to

same extent and is often just added as an effect and without any plausible reference to the virtual scene. The method of auralization with real-time performance can be integrated into the technology of "virtual reality." The process of generating the cues for the respective senses (3D image, 3D audio, etc.) is called "rendering." Apparently, simple scenes of interaction, for instance, when a person is leaving a room and closes a door, require complex models of room acoustics and sound insulation. Otherwise, it is likely that coloration, loudness, and timbre of sound within and between the rooms are not sufficiently represented. Another example is the interactive movement of a sounding object behind a barrier or inside an opening of a structure, so that the object is no longer visible but can be

heard by diffraction.

### *1.1. Sound field modeling*

The task of producing a realistic acoustic perception, localization, and identification is a big challenge. In contrast to the visual representation, acoustics deal with a frequency range involving three orders of magnitude (20 Hz to 20 kHz and wavelengths from about 20 m to 2 cm). Neither approximations of small wavelengths nor large wavelengths can be assumed with general validity. Different physical laws, that is, diffraction at low frequencies, scattering at high frequencies, and specular reflections have to be applied to generate a physically based sound field modeling. Hence, from the physical point of view (this means, not to mention the challenge of implementation), the question of modeling and simulation of an exact virtual sound is by orders of magnitude more difficult than the task to create visual images. This might be the reason for the delayed implementation of acoustic components in virtual environments.

At present, personal computers are just capable of simulating plausible acoustical effects in real time. To reach this goal, numerous approximations will still have to be made. The ultimate aim for the resulting sound is not to be physically absolutely correct, but perceptually plausible. Knowledge about human sound perception is, therefore, a very important prerequisite for evaluating auralized sounds.

Cognition of the environment itself, external events, and—very important—a feedback of one’s own actions are supported by the hearing event. Especially in VR environments, the user’s immersion into the computer-generated scenery is a very important aspect. In that sense, immersion can be defined as addressing all human sensory subsystems in a natural way. As recipients, humans evaluate the diverse characteristics of the total sound segregated into the individual objects. Furthermore, they evaluate the environment itself, its size, and the mean absorption (state of furniture or fitting). In the case of an acoustic scene in a room, which is probably typical for the majority of VR applications, a physically adequate representation of all these subjective impressions must, therefore, be simulated, auralized, and reproduced. Plausibility can, however, only be defined for specific environments. Therefore, a general approach of sound field modeling requires a physical basis and applicability in a wide range of rooms, buildings, or outdoor environments.

### *1.2. Reproduction*

The aural component additionally enforces the user’s immersive experience due to the comprehension of the environment through a spatial representation. Besides the sound field modeling itself, an adequate reproduction of the signals is very important. The goal is to transport all spatial cues contained in the signal in an aurally correct way to the ears of a listener. As mentioned above, coloration, loudness, and timbre are essential, but also the direction of a sound and its reflections are required for an at least plausible scene representation. The directional information in a spatial signal is very important to represent a room in its full complexity. In addition, this is supported by a dynamically adapted binaural rendering which enables the listener to move and turn within the generated virtual world.

### *1.3. System*

In this contribution, we describe the physical algorithmic approach of sound field modeling and 3D sound reproduction of the VR systems installed at RWTH Aachen University (see Figure 1). The system is implemented in a first version. It is open to any extended physical sound field modeling in real time, and is independent of any particular visual VR display technology, for example, CAVE-like displays or desktop-based solutions. Our 3D audio system named VirKopf has been implemented at the Institute of Technical Acoustics (ITA), RWTH Aachen University, as a distributed architecture. For any room acoustical simulation, VirKopf uses the software RAVEN (room acoustics for virtual environments) as a networked service (see Section 2.1). It is obvious that video and audio processing take a lot of computing resources for each subsystem, and by today’s standards, it is unrealistic to do all processing on a single machine. For that reason, the audio system realizes the computation of video and audio data on dedicated machines that are interconnected by a network. This idea is obvious and has already been successfully implemented by [4] or [5]. There are even commercially available solutions, which have been realized by dedicated hardware that can be used via a network interface, for example, the Lake HURON machine [6]. Other examples of acoustic rendering components that are bound by a networked interface can be found in connection with the DIVA project [7, 8] or Funkhouser's beam tracing approac[9]. Other approaches such as [2] or [10] have not been implemented as a networked client-server architecture but rely on a special hardware setup.

The VirKopf system differs from these approaches in some respects. A major difference is the focus of the VirKopf system, offering the possibility of a binaural sound experience for a moving listener without any need for headphones in immersive VR environments. Secondly, it is not implemented on top of any constrained hardware requirements such as the presence of specific DSP technology for audio processing. The VirKopf system realizes a software-only approach and can be used on off-the-shelf custom PC hardware. In addition to that, the system does not depend on specially positioned loudspeakers or a large number of loudspeakers. Four loudspeakers are sufficient to create a surrounding acoustic virtual environment for a single user using the binaural approach.

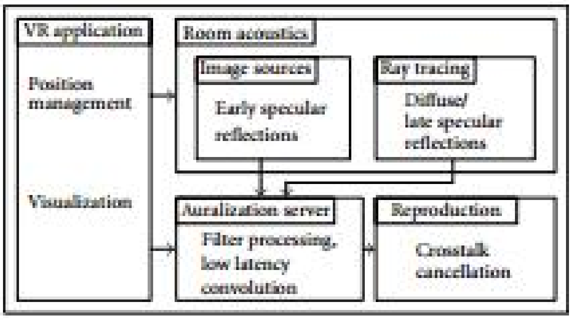


Figure 1: System components.

## 2. ROOM ACOUSTICAL SIMULATION

Due to several reasons, which cannot be explained in all details here, geometrical acoustics is the most important model used for auralization in room acoustics [11]. Wave models would be more exact, but only the approximations of geometrical acoustics and the corresponding algorithms provide a chance to simulate room impulse responses in real-time application. In this interpretation, delay line models, radiosity, or others are considered as basically geometric as well since wave propagation is reduced to the time-domain approach of energy transition from wall to wall. In geometrical acoustics, deterministic and stochastic methods are available. All deterministic simulation models used today are based on the physical model of image sources [12, 13]. They differ in the way how sound paths are identified by using forward (ray) tracing or reverse construction. Variants of this type of algorithms

are hybrid ray tracing, beam tracing, pyramid tracing, and so forth [14–20]. Impulse responses from image-like models consist of filtered Dirac pulses arranged accordingly to their delay and amplitude and are sampled with a certain temporal resolution. In intercomparisons of simulation programs[21, 22], it soon became clear that pure image source modeling would create too rough an approximation of physical sound fields in rooms since a very important aspect of room acoustics—surface and obstacle scattering—is neglected.

It can be shown that, from reflections of order two or three, scattering becomes a dominant effect in the temporal development of the room impulse response [23] even in rooms with rather smooth surfaces (see Figure 2). Fortunately, the particular directional distribution of scattered sound is irrelevant after the second or third reflection order and can well be assumed as Lambert scattering. However, in special cases of rooms with high

absorption such as recording studios, where directional diffusion coefficients are relevant, different scattering models have to be used. Solutions for the problem of surface scattering are given by either stochastic ray tracing or radiosity [14, 18, 24–27]. Furthermore, the fact that image sources are a good approximation for perfectly reflecting or low absorption surfaces is often forgotten. The approximation of images, however, is valid in large rooms at least for large distances between the source, wall, and receiver [28]. Another effect of wave physics—diffraction—can be introduced into geometrical acoustics[29, 30], but so far the online simulation has been restricted to stationary sound sources. Major problems arise, however, when extending diffraction models to higher orders. Apart
from outdoor applications, diffraction has not yet been implemented in the case of applications such as room acoustics. It should, however, be mentioned that numerous algorithmic details have already been published in the field of sound field rendering so far. New algorithmic schemes such as those presented by [31] have not yet been implemented. It should be kept in mind here that the two basic physical methods—deterministic sound images and stochastic scattering—should be taken into account in a sound field model with a certain performance of realistic physical behavior. Sound transmission as well as diffraction must be implemented in the cases of coupled rooms, in corridors, or cases where sound is transmitted through apertures.

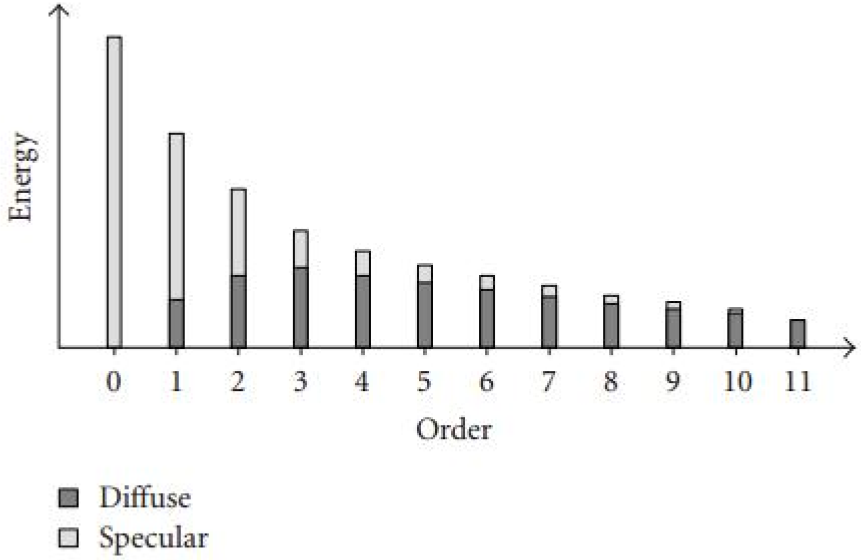


Figure 2: Conversion of specularly into diffusely reflected sound energy, illustrated by an example (after Kuttruff [23]).

### *2.1. Real-time capable implementation*

Any room acoustical simulation should take into account the above-mentioned physical aspects of sounds in rooms. Typically, software is available for calculating room impulse responses of a static source and a listener’s position within a few seconds or minutes. However, an unrestricted movement
of the receiver and the sound sources within the geometrical and physical boundaries are basic demands for any interactive on-line auralization. Furthermore, any interaction with the scenery, for instance, opening a door to a neighboring room, and the on-line-update of the change of the rooms’
modal structures should be provided by the simulation to produce a high believability of the virtual world [32]. At present, a room acoustical simulation software called RAVEN is being developed at our institute. The software aims at satisfying all above-mentioned criteria for a realistic simulation of the aural component, however, in respect of real-time capability. Special implementations offering the
possibility of room acoustical simulation in real time will be described in the following sections. RAVEN is basically an upgrade and enhancement of the hybrid room acoustical simulation method by Vorlander [20],

which was further extended by Heinz [25]. A very flexible and fast-toaccess framework for processing an arbitrary number of rooms (see Section 2.2) has been incorporated to gain a high
level of interactivity for the simulation and to achieve realtime capability for algorithms under certain constraints (see Section 5.2). Image sources are used for determining early reflections (see Section 2.3) in order to provide a most accurate localization of primary sound sources (precedence effect [33]) during the simulation. Scattering and reverberation are estimated on-line by means of an improved stochastic ray tracing method, which will be further described in
Section 2.4.

### *2.2. Scene partitioning*

The determination of the rooms' sound reflections requires an enormous number of intersection tests between rays and the rooms' geometry since geometrical acoustics methods treat sound waves as "light" rays. To apply these methods in real time, data structures are required for an efficient representation and determination of spatial relationships between sound rays and the room geometry.

These data structures organize geometry hierarchically in some $n$-dimensional space and are usually of recursive nature to accelerate remarkably queries of operations such as culling algorithms, intersection tests, or collision detections. Our auralization framework contains a preprocessing phase which transforms every single room geometry into a flexible data structure by using binary space partitioning
(BSP) trees for fast intersection tests during the simulation. Furthermore, the concept of scene graphs, which is basically a logical layer on top of the single room data structures, is used to make this framework applicable for an arbitrary number of rooms and to acquire a high level of interactivity for the room acoustical simulation.

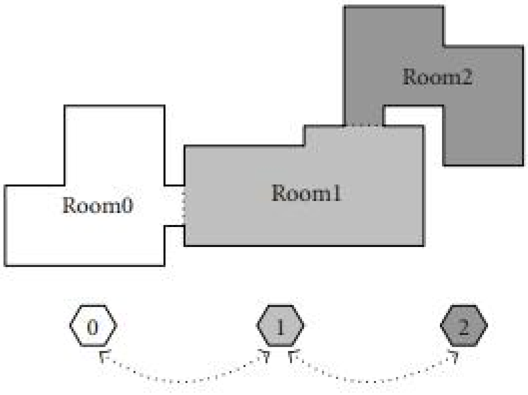


Figure 3: The scenery is split into three rooms, which are represented by the nodes of the scene graph (denoted through hexagons). The rooms are connected to their neighboring rooms by 2 portals (room0/room1 and room1/room2, denoted through the dotted
lines).

#### *2.2.1. Scene graph architecture*

To achieve an efficient data handling for an arbitrary number of rooms, the concept of scene graphs has been used. A scene graph is a collection of nodes which are linked according to room adjacencies.

A node contains the logical and spatial representation of the corresponding subscene. Every node is linked to its neighbors by so-called portals, which represent entities connecting the respective rooms, for example, a door or a window (see Figure 3). It should be noted that the number of portals for a single node is not restricted, hence the scenery can be partitioned quite flexibly into subscenes. The great advantage of using portals is their binary nature as two states can occur. The state “active” connects two nodes defined by the portal, whereas the state “passive” cuts off the specific link. This provides a high level of interactivity for the room acoustical simulations as room neighborhoods can be changed on-line, for instance, doors may be opened or closed. In addition, information about portal states can be exploited to speed up any required tests during the on-line room acoustical simulation by neglecting rooms which are acoustically not of interest, for example, rooms that are out of bounds for the current receiver’s position.

### ***2.3. Image source method***

The concept of the traditional image source (IS) method provides a quite flexible data structure, as, for instance, the online movement of primary sound sources and their corresponding image sources is supported and can be updated within milliseconds. Unfortunately, the method fails to simulate large sceneries as the computational costs are dominated by the exponential growth of image sources with an increasing number of rooms, that is, polygons and reflection order. Applying the IS method to an arbitrary number of rooms would result in an explosion of IS to be processed, which would make a simulation of a large virtual environ-ment impossible within real-time constraints due to the extreme number of IS to be tested online on audibility.

However, the scene graph data structure (see Section 2.2.1) provides the possibility of precomputing subsets of potentially audible IS according to the current portal configuration by sorting the entire set of IS dependent on the room(s) they originate from. This can easily be done by preprocessing the power set of the scene *S*, where *S* is a set of *n* rooms. The power set of *S* contains $2_n$ elements, and every subset, that is, family set of *S* refers to an *n*-bit number, where the *m*th bit refers to activity or inactivity of the *m*th room of *S*. Then, all ISs are sorted into the respective family sets of *S* by gathering information about the room IDs of the planes they have been mirrored on. Figure 5 shows exemplarily the power set *P* of a scenery *S* containing the three rooms *R*2, *R*1, *R*0, and the linked subsets of IS, that is, *P*(*S*) *={{*Primary Source*}*,*{*IS(*R*0)*}*,*{*IS(*R*1)*}*,*{*IS(*R*1, *R*2)*}*,*{*IS(*R*2)*}*,*{*IS(*R*2, *R*0)*}*, *{*IS(*R*2, *R*1)*}*, *{*IS(*R*2, *R*1, *R*0)*}}*.

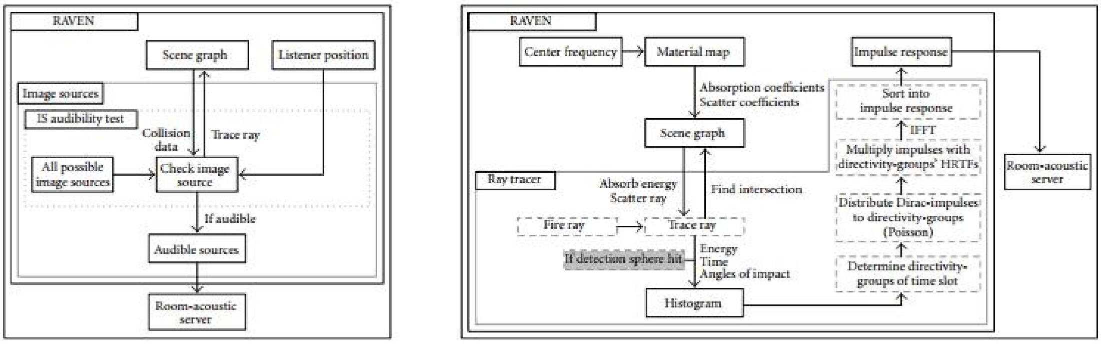
During on-line auralization, a depth-first search of the scene graph determines reachable room IDs for the current receiver’s position. This excludes both rooms that are out of bounds and rooms that are blocked by portals. This set of room IDs is encoded by the power set *P* to set unreachable rooms invalid as they are acoustically not of interest. If in the case of this example room *R*2 gets unreachable for the current receiver’s position, for example, someone closed the door, only IS family sets of *P* have to be processed for auralization that do not contain the room ID *R*2. As a consequence thereof, the number of IS family sets to be tested on audibility drops from eight to four, that is, *P*(0), *P*(1), *P*(2), *P*(3), which obviously leads to a significant reduction of computation time.

During simulation it will have to be checked whether every possible audible image source, which is determined as described above, is audible for the current receiver’s position (see Figure 4(a)). Taking great advantage of the scene graph’s underlying BSP-tree structures and an efficient tree traversing strategy, the required IS audibility test can be done very fast (performance issues are discussed in more detail in Section 5.2.1). If an image source is tested on audibility for the current receiver’s position, all data being required for filter calculation (position, intersection points, and hit material) will be stored in the super-ordinated container “audible sources” (see Figure 4(a)).

### ***2.4. Ray tracing***

The computation of the diffuse sound field is based on the stochastic ray tracing algorithm proposed by Heinz.

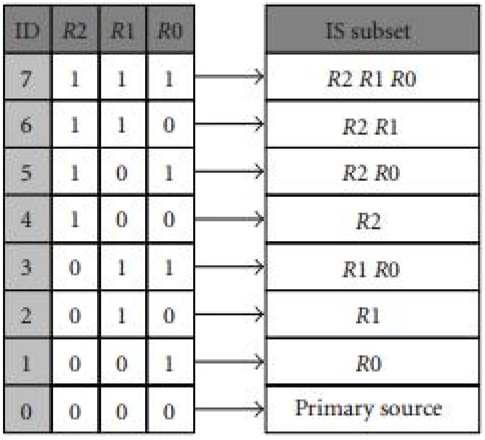
For building the binaural impulse response from the ray tracing data, Heinz assumed that the reverberation is ideally diffuse. This assumption is, however, too rough, if the room geometry is extremely long or flat and if it contains objects like columns or privacy screens. Room acoustical defects such as (flutter) echos would remain undetected [40, 41]. For a more realistic room acoustical simulation, the algorithm has been changed in a way so that these effects are taken into account (see Figure 4(b)).



(a) Image sources

(b) Ray tracing

Figure 4: (a) Image source audibility test, (b) estimation of scattering and reverberation.



| ID | *R2* | *R1* | *R0* | IS subset |
|---|---|---|---|---|
| 7 | 1 | 1 | 1 | *R2 R1 R0* |
| 6 | 1 | 1 | 0 | *R2 R1* |
| 5 | 1 | 0 | 1 | *R2 R0* |
| 4 | 1 | 0 | 0 | *R2* |
| 3 | 0 | 1 | 1 | *R1 R0* |
| 2 | 0 | 1 | 0 | *R1* |
| 1 | 0 | 0 | 1 | *R0* |
| 0 | 0 | 0 | 0 | Primary source |

Figure 5: IS/room-combination-power set $P(S)$ for a three-room situation. All IS are sorted into encapsulated containers depending on the room combination they have been generated from.

This aspect is an innovation in real-time virtual acoustics, which is to be considered as an important extension of the perceptive dimension.

The BSP-based ray tracing simulation starts by emitting a finite number of particles from each sound source at random angles where each particle carries a source directivity dependent amount of energy. Every particle loses energy while propagating due to air absorption and occurring reflections on walls, either specular or diffuse, and other geometric objects inside the rooms, that is, a material dependent absorption of sound. The particle gets terminated as soon as the particle's energy is reduced under a predefined threshold. Before a time $t_0$, which represents the image source cut-off time, only particles are detected which have been reflected specular with a diffuse history in order to preserve a correct energy balance. After $t_0$, all possible permutations of reflection types are processed (e.g., diffuse, specular, diffuse, diffuse, etc.).

The ray tracing is performed for each frequency band due to frequency dependent absorption and scattering coefficients, which results in a three-dimensional data container called histogram. This histogram is considered as the temporal envelope of the energetic spatial impulse response. One

single field of the histogram contains information about rays (their energy on arrival, time, and angles of impact) which hit the detection sphere during a time interval $\Delta t$ for a discrete frequency interval $f_b$. At first, the mean energy for fields with different frequencies but the same time interval is calculated to obtain the short-time energy spectral density. This step is also used to create a ray directivity distribution over time for the respective rays: for each time slot, the detection sphere is divided into evenly distributed partitions, so-called directivity groups. If a ray hits the sphere, the ray's remaining energy on impact is added to the corresponding sphere's directivity group depending on its time and direction of arrival (see Figure 6).

This energy distribution is used to determine a ray probability for each directivity group and each time interval $\Delta t$. Then a Poisson process with a rate equal to the rate of reflections for the given room and the given time interval is created. Each impulse of the process is allotted to the respective directivity group depending on the determined ray probability distribution. In a final step, each directivity group which was hit by a Poisson impulse cluster is multiplied by its respective HRTF, superposed to a binaural signal, and weighted by the square root of the energy spectral density. After that, the signal is transformed into time domain. This is done for every time step of the histogram and put together to the complete binaural impulse response. The ray tracing algorithm is managed by the room acoustics server to provide the possibility of a dynamic update depth for determining the diffuse sound field component (see Section 3). Since this contribution focuses on the implementation and performance of the complete system, no further details are presented here. A detailed description of the fast implementation and test results can be found in.

## 3. FILTER PROCESSING

For a dynamic auralization where the listener is allowed to move, turn, and interact with the presented scenery and where the sources can also be moved, the room impulse response has to be updated very fast. This becomes also more important in combination with congruent video images. Thus, the filter processing is a crucial part of the realtime process. The whole filter construction is separated into two parts. The most important section of a binaural room impulse response is the first part containing the direct sound and the early reflections of the room. These early reflections are represented by the calculated image sources and have to be updated at a rate which has to be sufficient for the binaural processing. For this reason, the operation interface between the room acoustics server and the auralization server is the list of the currently audible sources. The second part of the room impulse response is calculated on the room acoustics server (or cluster) to minimize the time required by the network transfer because the amount of data required to calculate the room impulse response is significantly higher than the resulting filter itself.

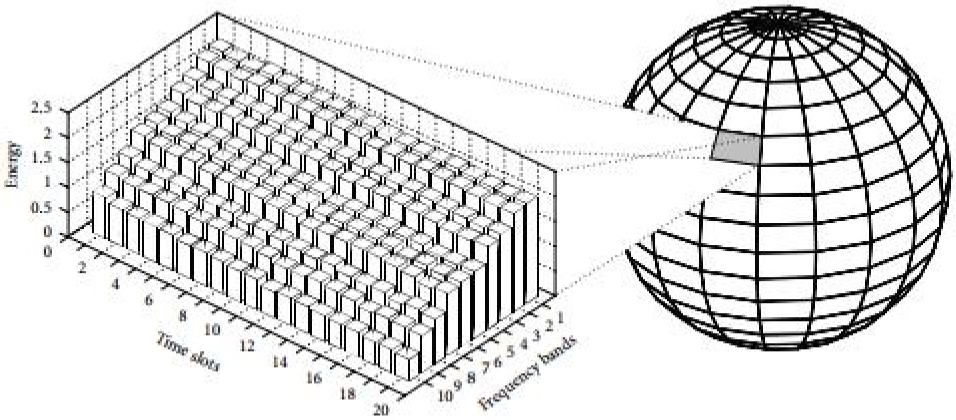


Figure 6: Histogram example of a single directivity group.

***3.1. Image sources***

Every single fraction of the complete impulse response, either the direct sound or the sound reflected by one or more walls, runs through several filter elements as shown in Figure 7. Elements such as directivity, wall, and air absorption are filters in a logarithmic frequency representation with a third octave band scale with 31 values from 20 Hz to 20 kHz. These filters contain no phase information so that only a single multiplication is needed. The drawback of using a logarithmic representation is the necessity of interpolation to multiply the resulting filter with the HRTF. But this is still not as computationally expensive as using a linear representation for all elements, particularly if more wall filters have to be considered for the specific reflection.

So far, the wall absorption filters are independent of the angle of sound incidence, which is a common assumption for room acoustical models. It can be extended to consider angle-dependent data if necessary. Reflections calculated by using the image source model will be attenuated by the factor

of the energy which is distributed by the diffuse reflections. The diffuse reflections will be handled by the ray tracing algorithm, (see Section 3.2).

Another important influence on the sound in a room, especially a large hall, is the directivity of the source. This is even more important for a dynamic auralization where not only the listener is allowed to move and interact with the scenery but where the sources can also move or turn. The naturalness of the whole generated sound scene is improved by every dynamic aspect being taken into account. The program accepts external directivity databases of any spatial resolution, and the internal database has a spatial resolution of 5 degrees for azimuth and elevation angles. This database contains the directivity of a singer and several natural instruments. Furthermore, it is possible to generate a directivity manually. The air absorption filter is only distance dependent and is applied also to the direct sound, which is essential for far distances between the listener and source.

At the end of every filter pass, which represents, up to now, a mono signal, an HRTF has to be used to generate a binaural head-related signal which contains all directional information. All HRTFs used by the VirKopf system were measured with the artificial head of the ITA for the full sphere due to the asymmetrical pinnae and head geometry. Nonsymmetrical pinnae lead to positive effects on the perceived externalization of the generated virtual sources. A strong impulse component such as the direct sound carries the most important spatial information of a source in a room. In order to provide a better resolution, even at low frequencies, an HRTF of a higher resolution is used for the direct sound. The FIR filter length is chosen to be 512 taps. Due to the fact that the filter processing is done in the frequency domain, the filter is represented by 257 complex frequency domain values corresponding to a linear resolution of 86 Hz.

Furthermore, the database does not only contain HRTFs measured at one specific distance but, also near-field HRTFs. This provides the possibility of simulating near-to-head sources in a natural way. Tests showed that the increasing interaural level difference (ILD) becomes audible at a distance

of 1.5 m or closer to the head. This test was performed in the semianechoic chamber of the ITA, examining the ranges where different near-field HRTFs have to be applied. The listeners were asked to compare signals from simulated HRTFs with those from correspondingly measured HRTFs on two criteria, namely, the perceived location of the source and any coloration of the signals. The simulated HRTFs were prepared from far-field HRTFs (measured at a distance of two meters) with a simple-level correction applied likewise to both channels. All of the nine listeners reported differences with regard to lateral sound incidences in the case of distances being closer than 1.5 m. No difference with regard to frontal sound incidences was reported in the case of distances being closer than 0.6 m. These results are very similar to the results obtained by research carried out in other labs, for example, [44]. Hence, HRTFs were measured at distances of 0.2 m, 0.3 m, 0.4 m, 0.5 m, 0.75 m, 1.0 m, 1.5 m, and 2.0 m. The spatial resolution of the databases is 1 degree for azimuth and 5 degrees for elevation angles for both the direct sound and the reflections.

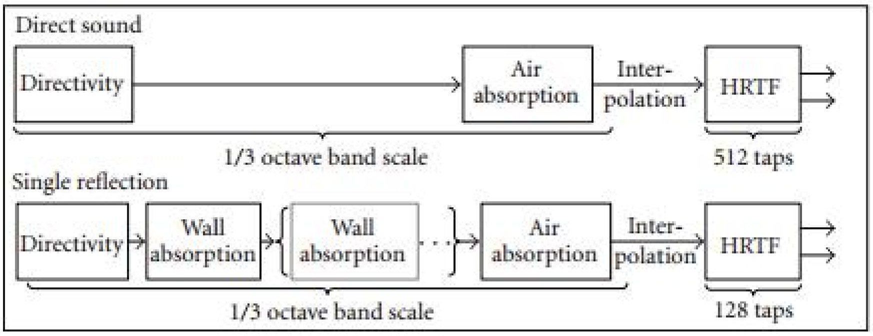


Figure 7: Filter elements for direct sound and reflections.

The FIR filter length of 128 taps used for the contribution of image sources is lower than for the direct sound, but is still higher than the limits to be found in literature. Investigations regarding the effects of a reduced filter length on localization can be found in [45]. As for the direct sound, the filter processing is done in the frequency domain with the corresponding filter representation of 65 complex values.

Using 128 FIR coefficients leads to the same localization results, but brings about a considerable reduction of the processing time (see Table 3). This was tested as well in internal listening experiences but is also congruent to the findings of other labs, that is, [46]. The spatial representation of image

sources is realized by using HRTFs measured in 2.0 m. In this case, this does not mean any simplification because the room acoustical simulation using image sources is not valid anyway at distances close (a few wavelengths) to a wall. A more detailed investigation relating to that topic can be found in.

### *3.2. Ray tracing*

As mentioned above, the calculation of the binaural impulse response of the ray tracing process is done on the ray tracing server in order to reduce the amount of data which has to be transferred via the network. To keep the filters up-to-date according to the importance of the filter segment, which is related to the time alignment, the auralization process can send interrupt commands to the simulation server. If a source or the listener is moving too fast to finish the calculation of the filter within an adequate time slot, the running ray tracing process will be stopped. This means that the update depth of the filter depends on the movements of the listener or the sources. In order to achieve an interruptible ray tracing process, it is necessary to divide the whole filter length into several parts. When a ray reaches the specified time stamp, the data necessary to restart the ray at this position will be saved and the next ray is calculated. After finishing the calculation of all rays, the filter will be processed up to the time the ray tracing updated the information in the histogram (this can also be a parallel process, if provided by the hardware). At this time, it is also possible to send the first updated filter section to the auralization server, which means that it is possible to take the earlier part of the changed impulse response into account before the complete ray tracing is finished. At this point, the ray tracing process will decide on the interrupt flag whether the calculation is restarted at the beginning of the filter or at the last time stamp. For slight or slow movements of the head or of the sources, the ray tracing process

has enough time to run through a complete calculation cycle containing all filter time segments. This also leads to the fact that the level of the simulation’s accuracy rises with the duration the listener stands at approximately the same position and the sources do not move.

## 4. REPRODUCTION SYSTEM

The primary reproduction system of the room acoustical modeling described in this paper is a setup mounted in the CAVE-like environment, which is a five-sided projection system of a rectangular shape, installed at RWTH

Aachen University. The special shape enables the use of the full resolution of 1600 by 1200 pixels of the LCD projectors on the walls and the floor as well as a 360 degree horizontal view. The dimensions of the projection volume are 3.60×2.70×2.70 m3 yielding a total projection screen area of 26.24 m2. Additionally, the use of passive stereo via circular polarization allows lightweight glasses. Head and interaction device tracking is realized by an optical tracking system. The setup of this display system is an improved implementation of the system that was developed with the clear aim to minimize attachments and encumbrances in order to improve user acceptance. In that sense, much of the credibility that CAVE-like environments earned in recent years has to be attributed to the fact that they try to be absolutely nonintrusive VR systems. As a consequence, a loudspeaker-based acoustical reproduction system seems to be the most desired solution for acoustical imaging in CAVE-like environments. Users should be able to step into the virtual scenery without too much preparation or calibration but still be immersed in a believable environment. For that reason, our CAVE-like environment depicted above was extended with a binaural reproduction system using loudspeakers.

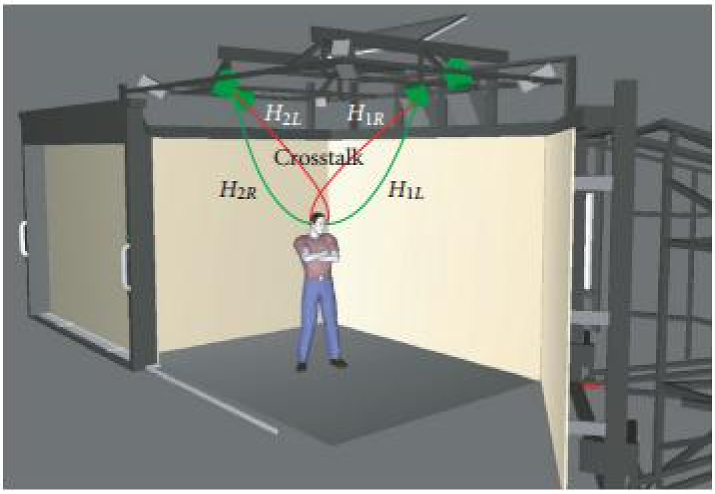


Figure 8: The CAVE-like environment at RWTH Aachen University. Four loudspeakers are mounted on the top rack of the system. The door, shown on the left, and a moveable wall, shown on the right, can be closed to allow a 360-degree view with no roof projection.

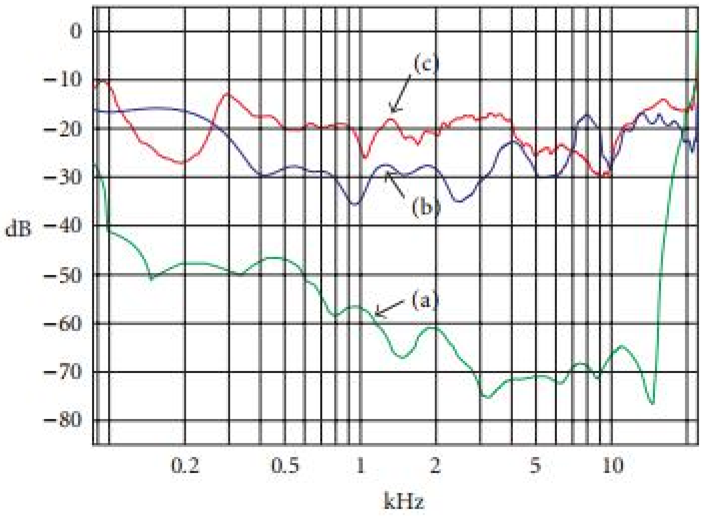


Figure 9: Measurement of the accessible channel separation using a filter length of 1024 taps. (a) = calculated, (b) = static solution, (c) = dynamic system.

### *4.1. Virtual headphone*

To reproduce the binaural signal at the ears with a sufficient channel separation without using headphones, a crosstalk cancellation (CTC) system is needed [49–51]. Doing the CTC work in an environment where the user should be able to walk around and turn his head requires a dynamic CTC system which is able to adapt during the listener's movements [52, 53]. The dynamic solution overrides the sweet spot limitation of a normal static crosstalk cancellation. Figure 8 shows the four transfer paths from the loudspeakers to the ears of the listener ($H_{1L}$ = transfer function loudspeaker 1 to left ear). A correct binaural reproduction means that the complete transfer function from the left input to the left ear (reference point is the entrance of the ear canal) including the transfer function $H_{1L}$ is meant to become a flat spectrum. The same is intended for the right transfer path, accordingly. The crosstalk indicated by $H_{1R}$ and $H_{2L}$ has to be canceled by the system.

Since the user of a virtual environment is already tracked to generate the correct stereoscopic video images, it is possible to calculate the CTC filter online for the current position and orientation of the user. The calculation at runtime enhances the flexibility of the VirKopf system regarding the validity area and the flexibility of the loudspeaker setup which can hardly be achieved with preprocessed filters. Thus, a database containing "all" possible HRTFs is required. The VirKopf system uses a database with a spatial resolution of one degree for both azimuth ($\phi$) and elevation ($\vartheta$). The HRTFs were measured at a frequency range of 100 Hz–20 kHz, allowing a cancellation in the same frequency range. It should be mentioned that a cancellation at higher frequencies is more error prone to misalignments of the loudspeakers and also to individual differences of the pinna. This is also shown by curve (c) in Figure 9. The distance between the loudspeaker and the head affects the time delay and the level of the signal. Using a database with HRTFs measured at a certain distance, these two parameters must be adjusted

by modifying the filter group delay and the level according to the spherical wave attenuation for the actual distance.

To provide a full head rotation of the user, a two loudspeaker setup will not be sufficient as the dynamic cancellation will only work in between the angle spanned by the loudspeakers. Thus, a dual CTC algorithm with a fourspeaker setup has been developed, which is further described in [54]. With four loudspeakers, eight combinations of a normal two-channel CTC system are possible and a proper cancellation can be achieved for every orientation of the listener. An angle dependent fading is used to change the active speakers in between the overlapping validity areas of two configurations.

Each time the head-tracker information is updated in the system, the deviation of the head to the position and orientation compared to the information given which caused the preceding filter change is calculated. Every degree of freedom is weighted with its own factor and then summed up.

Thus, the threshold can be parameterized in six degrees of freedom, positional values ($\Delta x$,$\Delta y$,$\Delta z$), and rotational values ($\Delta\phi$,$\Delta\vartheta$,$\Delta\rho$). A filter update will be performed when the weighted sum is above 1. The lateral movement and the head rotation in the horizontal plane are most critical so $\Delta x = \Delta y =$ 1cm and $\Delta\phi = 1.0$ degree are chosen to dominate the filter update. The threshold always refers to the value

where the limit was exceeded the last time. The resulting hysteresis prevents a permanent switching between two filters as it may occur when a fixed spacing determines the boundaries between two filters and the tracking data jitter slightly.

One of the fundamental requirements of the sound output device is that the channels work absolutely synchronously. Otherwise, the calculated crosstalk paths do not fit with the given condition. On this account, the special audio protocol ASIO designed by Steinberg for professional audio recording was chosen to address the output device [55].

To classify the performance that could be reached theoretically by the dynamic system, measurements of a static system were made to have a realistic reference for the achieved channel separation. Under absolute ideal circumstances, the HRTFs used to calculate the crosstalk cancellation filters are the same as during reproduction (individual HRTFs of the listener). In a first test, the crosstalk cancellation filters were processed with HRTFs of an artificial head in a fixed position. The windowing to a certain filter length and the smoothing give rise to a limitation of the channel separation. The internal filter calculation length is chosen to 2048 taps in order to take into account the time offsets caused by the distance to the speakers. The HRTFs were smoothed with a bandwidth of 1/6 octave to reduce the small dips which may cause problems by inverting the filters. After the calculation, the filter set is truncated to the final filter length of 1024 taps, the same length that the dynamic system works with. However,

the time alignment among the single filters is not affected by the truncation. The calculated channel separation using this (truncated) filter set and the smoothed HRTFs as reference is plotted in Figure 9 curve (a). Thereafter, the achieved channel separation was measured at the ears of the artificial head, which had not been moved since the HRTF measurement (Figure 9 curve (b)).

In comparison to the ideal reference cases, Figure 9 curve(c) shows the achieved channel separation of the dynamic CTC system. The main difference between the static and the dynamic system is the set of HRTFs used for filter calculation. The dynamic system has to choose the appropriate HRTF from a database and has to adjust the delay and the level depending on the position data. All these adjustments

cause minor deviations from the ideal HRTF measured directly at this point. For this reason, the channel separation of the dynamic system is not as high as the one that can be achieved by a system with direct HRTF measurement.

The theory of crosstalk cancellation is based on the assumption of a reproduction in an anechoic environment. However, the projection walls of CAVE-like environments consist of solid material causing reflections that decrease the performance of the CTC system. Listening tests with our system show that the subjective localization performance is still remarkably good. Also tests of other labs

[57, 58] and different CTC systems indicate a better subjective performance than it would be expected from measurements. One aspect validating this phenomenon is the precedence effect by which sound localization is primarily determined by the first arriving wavefront; the other aspect is the head movement which gives the user the ability to approve the perceived direction of incidence. A more

detailed investigation on the performance of our binaural rendering and reproduction system can be found in.

The latency of the audio reproduction system is the time elapsed between the update of a new position and orientation of the listener, and the point in time at which the output signal is generated with the recalculated filters. The output block length of the convolution (overlap save) is 256 taps as well as the chosen buffer length of the sound output device, resulting in a time between two buffer switches of

5.8 milliseconds at 44.1 kHz sampling rate for the rendering of a single block. The calculation of a new CTC filter set (1024 taps) takes 3.5 milliseconds on our test system. In a worst case scenario, the filter calculation just finishes after the sound output device fetched the next block, so it takes the time playing this block until the updated filter becomes active at the output. That would cause a latency of one block. In such a case, the overall latency accumulates to 9.3 milliseconds.

### *4.2. Low-latency convolution*

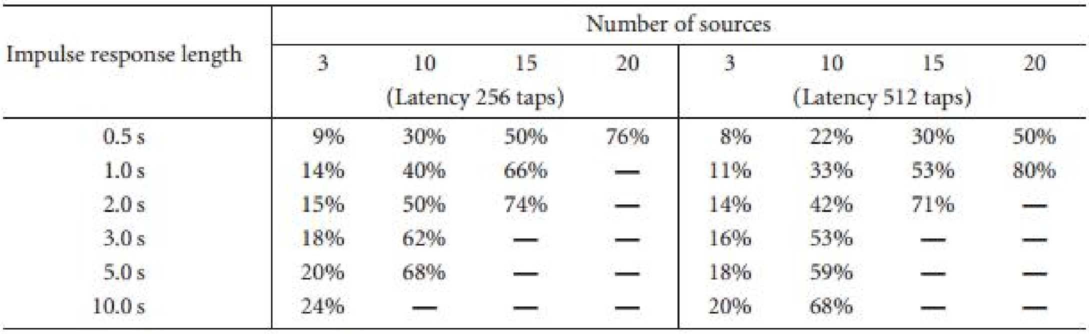
A part of the complete dynamic auralization system requiring a high amount of processing power is the convolution of the audio signal. A pure FIR filtering would cause no additional latency except for the delay of the first impulse of the filter, but it also causes the highest amount of processing power. Impulse responses of more than 100 000 taps or more cannot be processed in real time on a PC system using FIR filters in the time domain. The block convolution is a method that reduces the computational cost to a minimum, but the latency increases in proportion to the filter length. The only way to minimize the latency of the convolution is a special conditioning of the complete impulse response in

filter blocks. Basically, we use an algorithm which works in the frequency domain with small block sizes at the beginning of the filter and increasing sizes to the end of the filter. More general details about these convolution techniques can be found in [60]. However, our algorithm does not operate on the commonly used segmentation which doubles the block length every other block. Our system provides a special block size conditioning with regard to the specific PC hardware properties as, for instance, cache size or special processing structures such as SIMD (single instruction multiple data). Hence, the optimal convolution adds a time delay of only the first block to the latency of the system, so

that it is recommended to use a block length as small as possible. The amount of processing power is not linear to the overall filter length and also constrained by the chosen start block length. Due to this, measurements were done to determine the processor load of different modes of operation (see

Table 1).

Table 1: CPU load of the low-latency convolution algorithm.



| Impulse response length | Number of sources | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 3 | 10 | 15 | 20 | 3 | 10 | 15 | 20 |
| | | (Latency 256 taps) | | | | (Latency 512 taps) | | |
| 0.5 s | 9% | 30% | 50% | 76% | 8% | 22% | 30% | 50% |
| 1.0 s | 14% | 40% | 66% | — | 11% | 33% | 53% | 80% |
| 2.0 s | 15% | 50% | 74% | — | 14% | 42% | 71% | — |
| 3.0 s | 18% | 62% | — | — | 16% | 53% | — | — |
| 5.0 s | 20% | 68% | — | — | 18% | 59% | — | — |
| 10.0 s | 24% | — | — | — | 20% | 68% | — | — |

## 5. SYSTEM INTEGRATION

The VirKopf system constitutes the binaural synthesis and reproduction system, the visual-acoustic coupling, and it is connected to the RAVEN system for room acoustical simulations. The complete system's layout with all components is shown in Figure 10. As such it describes the distributed

system which is used for auralization in the CAVE-like environment at RWTH Aachen University, where user interaction is tracked by six cameras. As a visual VR machine, a dual Pentium 4 machine with 3 GHz CPU speed and 2 GB of RAM is used (cluster master). The host for the audio VR subsystem is a dual Opteron machine with 2 GHz CPU speed and 1 GB of RAM. The room acoustical simulations run on Athlon 3000+ machines with 2 GB of RAM. This hardware

configuration is also used as a test system for all performance measurements. As audio hardware, an RME Hammerfall system is used which allows sound output streaming with a scalable buffer size and a minimum latency of 1.5 milliseconds. In our case, an output buffer size is chosen to 256 taps (5.8 milliseconds). The network interconnection between all PCs was a standard Gigabit Ethernet.

### *5.1. Real-time requirements*

Central aspects of coupled real-time systems are latency and the update rate for the communication. In order to get an objective criterion for the required update rates, it is mandatory to inspect typical behavior inside CAVE-like environments with special respect to head movement types and magnitude

of position or velocity changes.

In general, user movements in CAVE-like environments can be classified in three categories [61]. One category is identified by the movement behavior of the user inspecting a fixed object by moving up and down and from one side to the other in order to accumulate information about its structural properties. A second category can be seen in the movements when the user is standing at one spot and uses head or body rotations to view different display surfaces of the CAVE. The third category for head movements can be observed when the user is doing both, walking and looking around in the CAVElike environment. Mainly, the typical applications we employ can be classified as instances of the last two categories, although the exact user movement profiles can be individually different. Theoretical and empirical discussions about typical head movement in virtual environments are still a subject of research, for example, see [61–63] or [64].

As a field study, we recorded tracking data of users' head movements while interacting in our virtual environment. From these data, we calculated the magnitude of the velocity of head rotation and translation in order to determine the requirements for the room acoustics simulation. Figure 11(a)

shows a histogram of the evaluated data for the translational velocity. Following from the deviation of the data, the mean translational velocity is at 15.4 cm/s, with a standard deviation of 15.8 cm/s and the data median at 10.2 cm/s, compare Figure 11(c). This indicates that the update rate of the room acoustical simulation can be rather low for translational movement as the overall sound impression does not change much in the immediate vicinity (see [65] for further information). As an example, imagine a room acoustical simulation of a concert hall where the threshold for triggering a recalculation of a raw room impulse response is 25 cm (which is typically half a seat row's distance). With respect to the translational movement profile of a user, a recalculation has to be done approximately every 750 milliseconds to catch about 70% of the movements. If the system aims at calculating correct image sources for about 90% of the movements, this will have to be done every 550 milliseconds. A raw impulse response contains the raw data of the images, their amplitude and delay, but not their direction in listener's coordinates. The slowly updated dataset represents, thus, the roomrelated cloud of image sources. The transformation into 3D listener's coordinates and the convolution will be updated much faster, certainly, in order to allow a direct and smooth

responsiveness.

CAVE-like environments allow the user to directly move in the scene, for example, by walking inside of the boundaries of the display surfaces and tracking area. Additionally, indirect navigation enables the user to move in the scenery virtually without moving his body but by pointing metaphors when using hand sensors or joysticks. Indirect navigation is mandatory, for example, for architectural walkthroughs as the virtual scenery is usually much larger than the space covered by the CAVE-like device itself. The maximum velocity for indirect navigations has to be limited in order to avoid artifacts or distortions in the acoustical rendering and perception. However, during the indirect movement, users do not tend to move their head and the overall sensation reduces the capability to evaluate the correctness of the simulation. Once the users stop, it takes about 750 milliseconds as depicted above to calculate the right filters for the current user position. We made the experience that a limitation of the maximum velocity for indirect navigation to 100 cm/s shows good results and user acceptance.

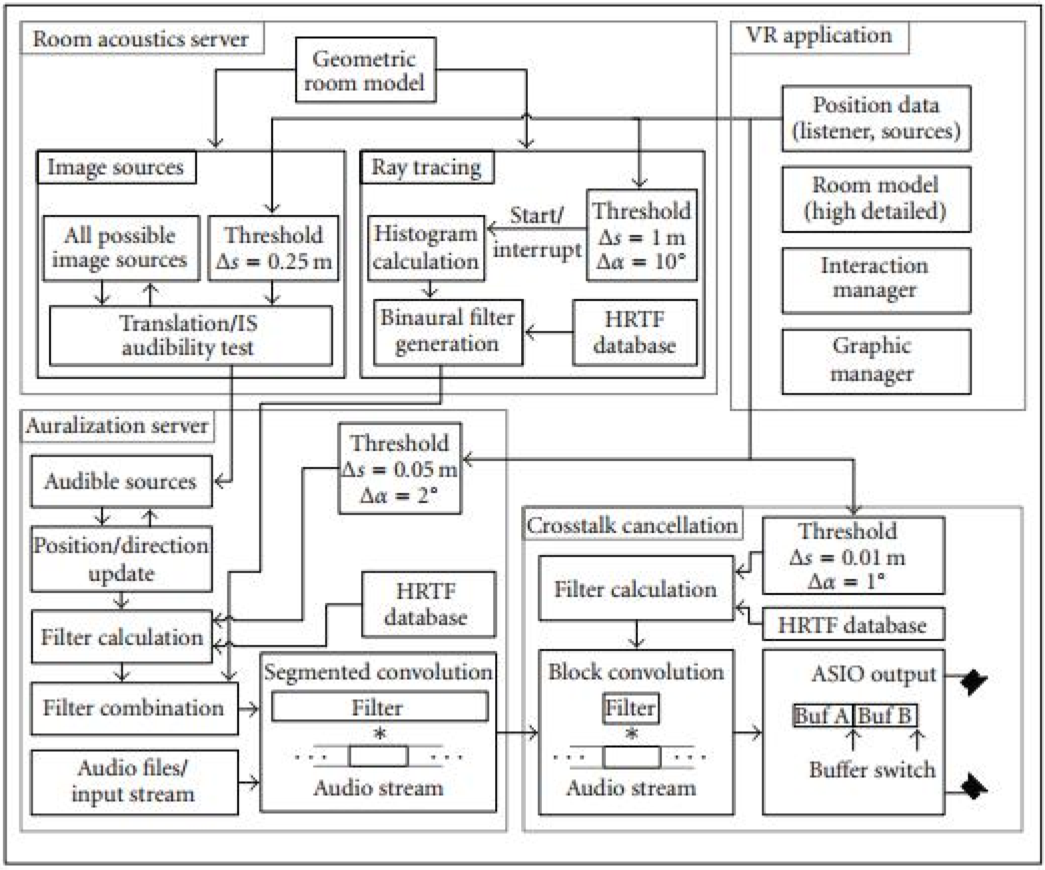
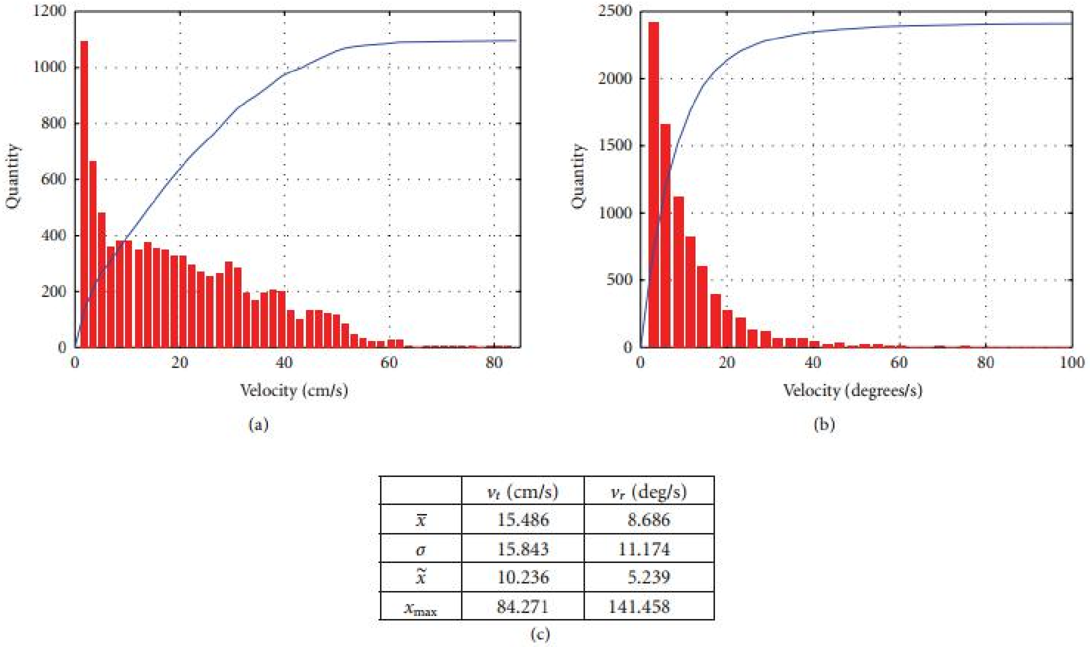


Figure 10: The complete binaural auralization system.



(a)

(b)

| | $v_t$ (cm/s) | $v_r$ (deg/s) |
|---|---|---|
| $\bar{x}$ | 15.486 | 8.686 |
| $\sigma$ | 15.843 | 11.174 |
| $\tilde{x}$ | 10.236 | 5.239 |
| $x_{\max}$ | 84.271 | 141.458 |

(c)

Figure 11: Histogram of translational ($v_t$) and rotational ($v_r$) velocities of movements of a user acting in a CAVE-like environment. The

blue line depicts the cumulative percentage of the measurements. In (b), we limited the upper bound to 100 degrees/s for better readability,

(c) shows the descriptive statistics about the measurements.

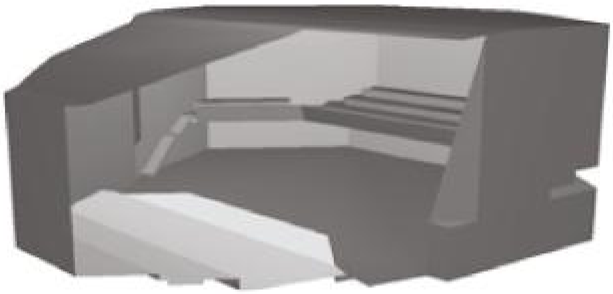


Figure 12: Sliced polygon model of the concert hall of Aachen’s Eurogress convention center.

In addition to the translational behavior, Figure 11(b) shows the rotational profile for head movements of a user. Peak angular velocities can be up to 140 degrees per second although these are very seldom. The mean for rotational movement is at 8.6 degrees/s with a standard deviation of 11.1 degrees/s and a data median at 5.2 degrees/s, compare Figure 11(c). Data sets provided as standard material for research on system latency, for example, by [66] or [61], show comparable results.

The orientation of the user’s head in the sound field is very critical as reflections have to be calculated for the headrelated impulse response in listener’s coordinates. The changing ITD of the HRTFs during head rotation may cause a significant phase mismatch of two filters. In cross-fading from one room impulse response to the next, these differences should not be too big as this might result in audible combfilter effects. To reduce these differences, a filter change every 1-2 degrees is necessary here. In order to be precise for almost all possible rotational velocities, we consider a timing interval for a recalculation every 10–20 milliseconds as mandatory. As a consequence, the block size configured in

the audio processing hardware should not be bigger than 512 samples as this limits the minimal possible update time to 11.6 milliseconds at a 44.1 kHz sampling rate.

### *5.2. Performance of the room acoustical simulation*

To evaluate the implementation and to determine its realtime capabilities, several experiments were carried out on the test system. For a realistic evaluation, a model of the concert hall of Aachen’s Eurogress (volume about 15 000 m3) convention center was constructed, which is shown in Figure 12.

All results presented in this contribution are based on this model.

The model is constructed of 105 polygons and 74 planes, respectively. Although it is kept quite simple, the model con-tains all indoor elements of the room which are acoustically of interest [67], for example, the stage, the skew wall elements, and the balustrade. Details of small elements are neglected and represented by equivalent scattering [68]. Surface properties, that is, absorption and scattering coefficients are defined through standardized material data [69, 70].

#### *5.2.1. Image source method performance*

The computation time for the translational movement of primary sound sources and their respective image sources depends solely on the number of image sources. An average computation time of about 1 millisecond per 1000 image sources was measured. The main part of the computation time is needed for the audibility test.

To give a better idea of the achieved speed up by the use of BSP trees, a brute-force IS audibility test has been implemented for comparison purpose. This algorithm tests every scene’s polygon on intersection instead of testing only a few room’s subpartitions by means of a BSP-tree structure. Figure 13 shows a comparison of

measured computation times for the IS-audibility test up to second IS order
of both approaches. As expected, the computation time of the brute-force method rises exponentially with the exponentially growing number of ISs, whereas the BSP-based approach has only a quite linearly growing computation time demand due to the drop of search complexity up to O(log *N*),
*N* number of polygons.

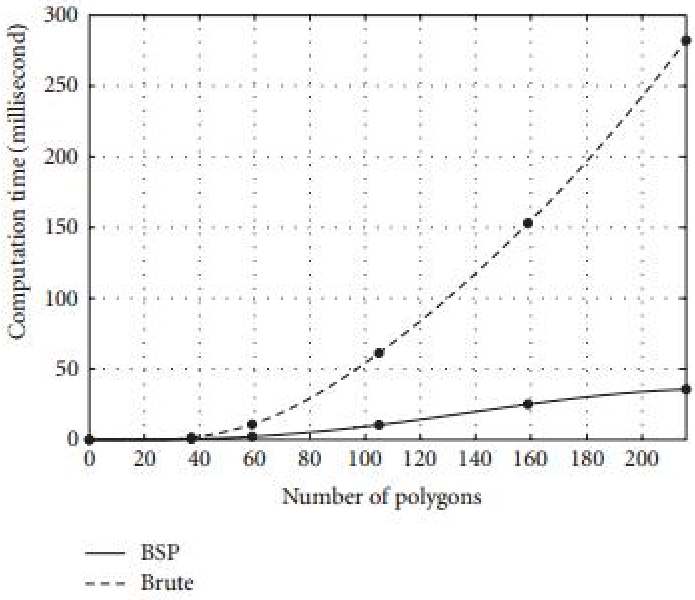
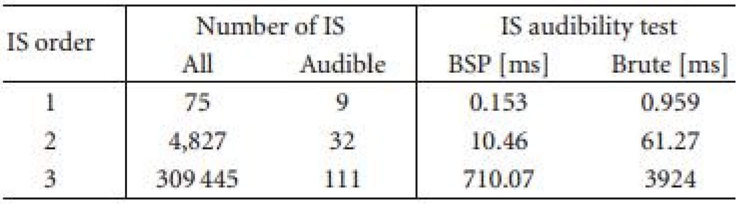


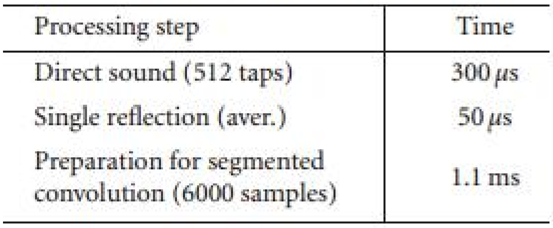
Figure 13: Comparison of required computation time for the ISs audibility test up to second-order ISs for different Eurogress models which differ in their level of detail (see [38] for details). With the growing number of polygons for the model's different levels of detail, the number of ISs grows exponentially, which leads to an exponential growth of the computation time for the bruteforce approach. The computation time demands of the BSP-based
method grows only linear due to the drop of search complexity up to O(log *N*), *N* number of polygons.

Table 2: Comparison of the measurement results of the IS audibility test.



| IS order | Number of IS | | IS audibility test | |
|---|---|---|---|---|
| | All | Audible | BSP [ms] | Brute [ms] |
| 1 | 75 | 9 | 0.153 | 0.959 |
| 2 | 4,827 | 32 | 10.46 | 61.27 |
| 3 | 309 445 | 111 | 710.07 | 3924 |

TABLE 3: Calculation time of several parts of the filter.



| Processing step | Time |
|---|---|
| Direct sound (512 taps) | 300 $\mu$s |
| Single reflection (aver.) | 50 $\mu$s |
| Preparation for segmented convolution (6000 samples) | 1.1 ms |

With the assigned time slot (see Section 5.1) of 750 milliseconds for the simulation process, real-time capability for a room acoustical simulation with all degrees of freedom such as movable sound sources, movable receiver, changing sources' directivities, and interaction with the scenery is reached for about 320 000 ISs to be tested during runtime. Applying these constraints to the measurement results of the
IS audibility test (see Table 2) makes the simulation of the Eurogress model real-time capable up to order 3.

Besides the performance of the room acoustical simulation, the processing time of the filter is very important. All time measurements of the calculation routines presented in this section are performed on our test system. Calculating the image sources of the Eurogress model up to the third order, 111 audible image sources can be found in the first part of the impulse response of 6000 samples length corresponding to 136 milliseconds. In this case, one source is placed on the stage, and the listener is located in the middle of the room. The complete filter processing (excluding the audibility test) is done in 6.95 milliseconds. Note, that the filter processing has different entry points. The rotation of the listener or a source does not cause a recalculation of the audible sources, only the filter has to be processed.

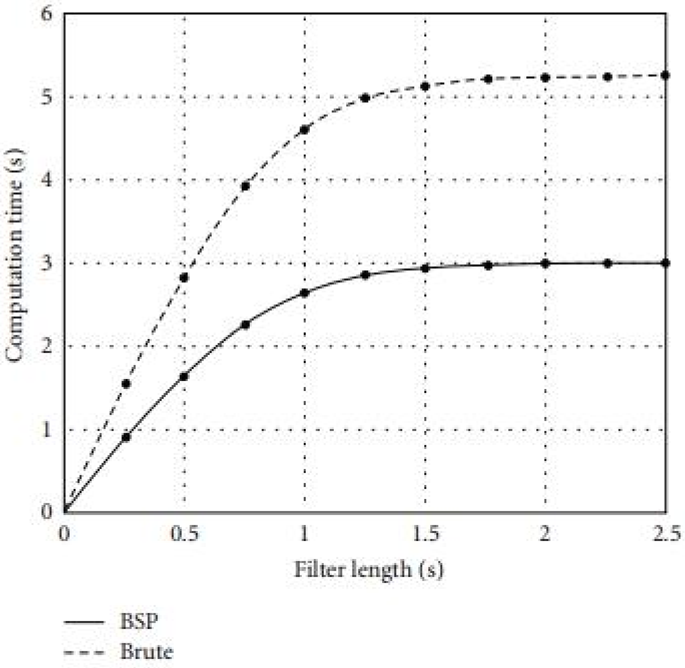


Figure 14: Comparison of required computation times for the determination of impulse responses with increasing length using 80 000 rays for the simulation.

*5.2.2. Ray-tracing performance*

For measuring the performance of the ray-tracing algorithm, all materials of the Eurogress model were replaced by a single one in order to avoid influences of different scattering and absorption coefficients on the results. As in the previous section, a brute-force ray tracing algorithm has been implemented to compare the results to the BSP-based method we use in our framework. While the brute-force approach has a linearly growing computation time, that is, a complexity of O(*N*), *N* number of polygons, the BSP-based algorithm grows only logarithmically with increasing time due to the drop of search complexity to O(log *N*) (see Figure 14, $t <$ 0.8second). A ray gets terminated if a minimum energy threshold is reached. Thus, both approaches get faster with increasing time due to the growing number of reflections, that is, the growing rays' loss of energy and ray termination, respectively. As an example, the algorithm needs an average of about 2.6 second per 80 000 rays (10 000 rays per frequency band, the first two octave bands are skipped) for determination of an impulse response with the length of 1 secone. As the processing time of the raytracing algorithm increases linearly with the number of rays used, a comparison of these results is redundant. It is obvious that the algorithm is able to cope with the real-time requirements, especially when using small numbers of rays at first to get a low-resolution histogram. If the listener stays at one place for a longer period of time, the ray tracer can update the histogram with more rays to get a higher resolution and determine a longer impulse response, respectively.

### *5.3. Network*

With respect to the timing, the optical tracking system is capable of delivering spatial updates of the position and orientation of the user’s head and an additional interaction device to the VR application in 18.91 milliseconds. This figure is a direct result from the sum of the time needed for the visual recognition of two tracking targets as well as the transmission time for the measured data over a network link. For applications that must have a minimum latency time and do not need wireless tracking, the usage of an electromagnetic tracking system can reduce the latency to ≈ 5milliseconds.

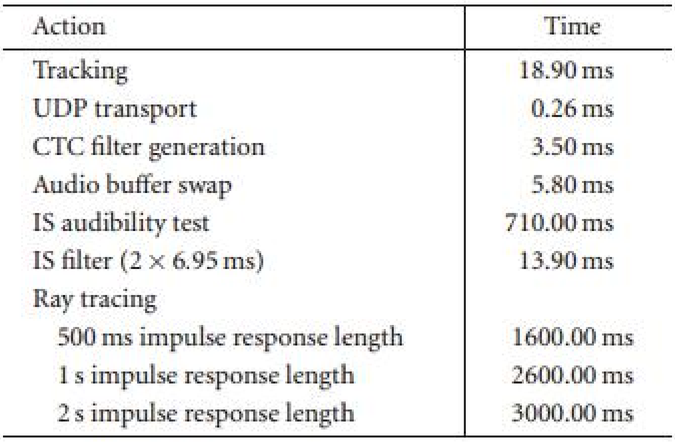
However, the VirKopf system distinguishes between two types of update messages. One type deals with low-frequency state changes such as commands to play or stop a specific sound. The second type updates the spatial attributes of the sound source and the listener at a high frequency. For the first type, a reliable transport protocol is used (TCP), while the latter is transmitted at a high frequency over a low overhead but possibly unreliable protocol (UDP).

In order to get an estimate of the costs of network transport, the largest possible TCP and UDP messages produced by the VirKopf system were transmitted from the VR application to the VirKopf server many times and then sent back. The transmission time for this round trip was taken and halved for a single-trip measurement. The worst case times of the single trips are taken as a basis for the estimation of the overall cost introduced by the network communication. The mean time for transmitting a TCP command was 0.15 millisecond±0.02 millisecond. The worst case transmission time on the TCP channel was close to 1.2 millisecond. UDP communication was measured for 20 000 spatial update tables for 25 sound sources, resulting in a transmission time for the table of 0.26 millisecond ± 0.01 millisecond. It seems surprising that UDP communication is more expensive than TCP, but this is a result from larger packet sizes of an spatial update (≈ 1 kB) in comparison to small TCP command sizes (≈ 150 bytes).

### *5.4. Overall performance*

Several aspects have to be taken into account to give an overview of the performance of the complete system, the performance of several subsystems, the organization of parallel processing, the network transport, but also of the scenery, namely, the simulated room (dimension and complexity of the geometry), the velocity of sources, and finally the user. Updating the room acoustical simulation is the most timeconsuming part of the system and requires a strategy of achieving the best perceptual performance. Image sources and ray tracing are processed independently on different CPUs. The binaural filter of the ray tracing process will be calculated directly on the ray-tracing server. The auralization server has to calculate the image source filter and combine all filter segments of the ray-tracing process. Figure 15 describes one possible segmentation of the ray tracing and combination of the image source filter. It should be mentioned that the length of the specular part is room dependent. The raytracing interrupt point will be adjusted based on the movement velocity of the listener and the sources. This means that the audio signal is filtered with the updated first part of the room impulse response while the generation of the late part by ray tracing is still in progress. The filter segment to be updated will be cut off from the complete filter with a short ramp of 32 samples ≈ 0.72 millisecond, and the new segment will be placed in with the same ramp to avoid audible artifacts.

Table 4: Overview of performance measurements of the several subsystems.



| Action | Time |
|---|---|
| Tracking | 18.90 ms |
| UDP transport | 0.26 ms |
| CTC filter generation | 3.50 ms |
| Audio buffer swap | 5.80 ms |
| IS audibility test | 710.00 ms |
| IS filter (2 × 6.95 ms) | 13.90 ms |
| Ray tracing | |
| 500 ms impulse response length | 1600.00 ms |
| 1 s impulse response length | 2600.00 ms |
| 2 s impulse response length | 3000.00 ms |

Due to the dependency of all these factors, update times cannot be estimated in general. For this reason, we will give some detailed examples with respect to the performance measurements (see Tables 4 and 5) made in several sections above. It should be noticed that the image source filter will be updated at any time the source or the head moved more than 2 cm or turned more than 1 degree, respectively. The image source filter will be calculated on the current list of audible sources (positions updated). The resulting filter only contains a few wrong reflections which will be removed after the audibility test. Thus, the specular reflections at the first part of the impulse response become audible with the correct spatial representation already after 35 milliseconds (tracking + UDP transport + CTC filter generation IS filter generation + audio buffer swap). This is also the time needed to react to a listener’s head rotation (see Table 5).

## 6. SUMMARY

In this contribution, we introduced a quite complex system for simulation and auralization of room acoustics in real time. The system is capable of simulating room acoustical sound fields in any kind of enclosures without the prerequisite of any diffuse-field conditions. The room shape can hence be extremely long, flat, coupled, or of any other special property. The surface properties, too, can be freely chosen by using the amount of wave scattering according to standardized material data. Furthermore, the system includes a sound field reproduction for a single user based on dynamic crosstalk cancellation (virtual headphone). The software is implemented on standard PC hardware and requires

no special processors. The performance (simulation processing time, filter update rates, tracker, and sound hardware latency) was evaluated and considered sufficiently in the case

of a concert hall of medium size.

Particular features of the system are the following.

(i) It is not based on any assumption of an ideal diffuse sound field but on a full room acoustic simulation in two parts. Specular and scattered components of the impulse response are treated separately. Any kind of room shape and volume can be processed except small rooms at low frequencies.

(ii) The decision with regard to the amount of specular and diffuse reflections is just room dependent and purely based on physical sound field aspects.

(iii) The user will just be involved to create the room CAD model and the standard material data of absorption and scattering. Therefore, import functions of commercial non-real-time simulation software can be used. The fact that the auralization is performed in

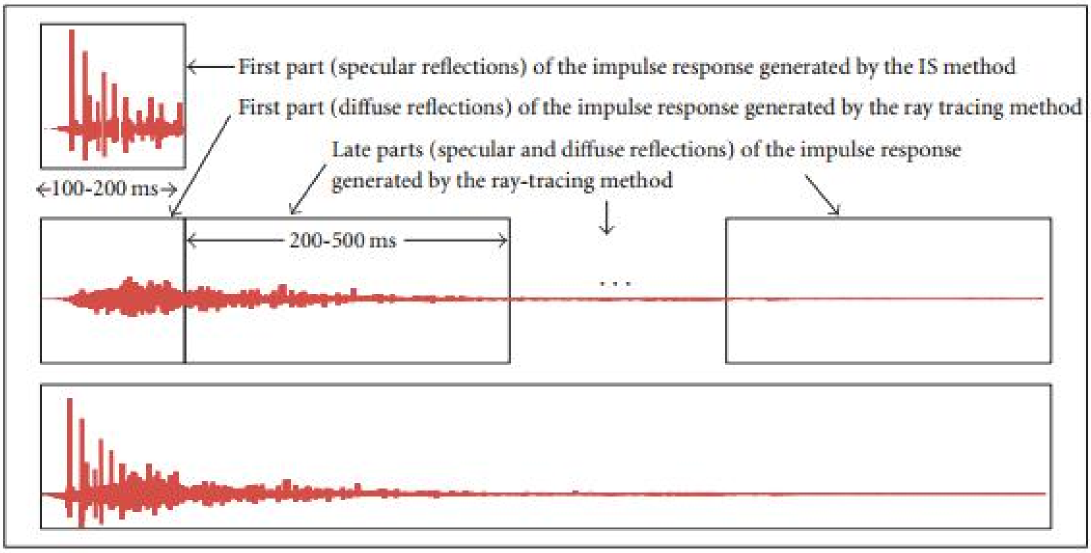
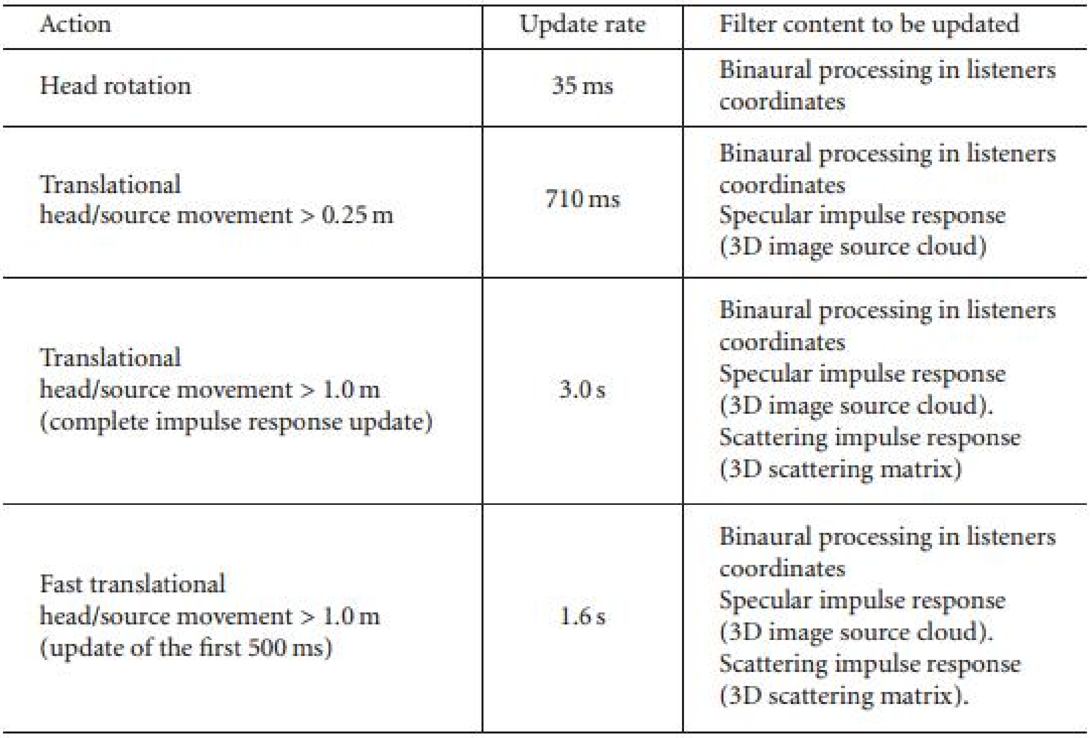

First part (specular reflections) of the impulse response generated by the IS method
First part (diffuse reflections) of the impulse response generated by the ray tracing method
Late parts (specular and diffuse reflections) of the impulse response generated by the ray-tracing method
←100-200 ms→
200-500 ms
. . .

Figure 15: Combination of filter (or filter segments) for one ear generated by ray tracing and the first part of the impulse response generated by the image source model.

Table 5: Update intervals for different modes and conditions of head or source movements based on the measurements shown in Table 4.



| Action | Update rate | Filter content to be updated |
|---|---|---|
| Head rotation | 35 ms | Binaural processing in listeners coordinates |
| Translational<br>head/source movement > 0.25 m | 710 ms | Binaural processing in listeners coordinates<br>Specular impulse response (3D image source cloud) |
| Translational<br>head/source movement > 1.0 m<br>(complete impulse response update) | 3.0 s | Binaural processing in listeners coordinates<br>Specular impulse response (3D image source cloud).<br>Scattering impulse response (3D scattering matrix) |
| Fast translational<br>head/source movement > 1.0 m<br>(update of the first 500 ms) | 1.6 s | Binaural processing in listeners coordinates<br>Specular impulse response (3D image source cloud).<br>Scattering impulse response (3D scattering matrix). |

real time means that the user is not required to carry out any additional tasks. The system will adjust all relevant runtime parameters automatically and inherently, like division into specular and scattered parts and filter update rates.

(iv) The treatment of the components of the binaural impulse response is separated regarding the simulation itself, the update rate to the auralization server, and the convolution process.

(v) The decision regarding the update rate and depth of impulse response simulation is based on the interaction and speed of movement of the user in the VR system.

(vi) The precision of details in the impulse response, its exactness of delays, and its exactness of direction of sound incidence are just depending on the relative arrival time in the impulse response. This is in agreement with the ability of the human hearing system regarding localization and echo delays. Is should also be mentioned here that the system parameters of simulation depth and update rate are not controlled by the user but inherently treated in the system. This way of processing will create full complexity and exact auralization in the very early part of the direct sound and the first reflections. Gradually, the sound energy will be transferred into the scattered component of the impulse. The precision and update rates are reduced, motivated by the limits due to psychoacoustic in masking effects. The system is open for further extension with respect to sound diffraction and sound insulation.

The real-time performance of the room acoustical simulation software was achieved by the introduction of a flexible framework for the interactive auralization of virtual environments. The concept of scene graphs for the efficient and flexible linkage of autonomously operating subscenes by

means of so-called portals has been incorporated into the existing framework and combined with an underlying BSP-tree structure for processing geometry issues very fast. The use of this framework provides the possibility of a significant reduction of computation time for both applied algorithms (deterministic image sources and a stochastic ray tracer). Especially, the image source method is improved by the introduction of spatial data structures as portal states can be exploited so that the number of image sources to be processed can be reduced remarkably.

A fast low latency engine ensures that impulse responses regardless of their complete length will be considered by the filtering of the mono audio material after 5.8 milliseconds (block length 256 samples). Optimizations concerning modern processor extensions enable the rendering of, for example, 10 sources with filters of 3-second (132 000 taps) length or 15 sources with filters of 2-second length.

The reproduction of the binaural audio signal is provided by a dynamic crosstalk cancellation system with no restrictions to user movements. This system acts as a virtual headphone providing the channel separation without the need to wear physical headphones.

Gigabit Ethernet is used to connect the visual rendering system and the audio system. The visual VR system transmits the control commands as well as the spatial updates of the head and the sources. The control commands (e.g., start/stop) will be considered in the audio server after 0.15 millisecond so that the changes are served with the next sound output block for a tight audio video synchronism.

## 7. OUTLOOK

Despite the good performance of the whole system, there are many aspects that have to be investigated. To further enhance the quality of the room acoustical simulation, physical effects like sound insulation and diffraction are to be incorporated into the existing algorithms. In addition, the simulation of frequencies below the Schroeder frequency could be done by means of a fast and dynamic finite element method (FEM)-solver. The existing framework is already open to take these phenomena into account, the respective algorithms have only to be implemented. At present, the simulation software is implemented in a first version as a self-contained stable base. Thus, optimizing the algorithms is necessary to further increase their performance, especially with focus on the computing of processes in parallel. Position prediction could be a possibility of reducing the deviation of the position, the filter was calculated for, and the actual listener's position.

Preliminary listening tests showed that the generated virtual sources could be localized at a low error-rate [59]. The room acoustical simulation was perceived as plausible and matching to the generated visual image. In the future, more tests will be accomplished to evaluate the limitation of the

update rates and the number of sources. Perception based reduction such as stated in, for example, [71, 72] is also an interesting method of reducing the processing costs, and will be considered in the future.

## ACKNOWLEDGMENTS

The authors would like to thank Frank Wefers, Hilmar Demuth, and Philipp Dross for their commitment during parts of the programming work, and also Torsten Kuhlen, Andreas Franck, and Mark-Oliver Guld for their support and discussion. Furthermore, thanks to the DFG for funding parts of the project (DFG-Project “The Virtual Headphone,” 2004–2006). Finally, the authors would like to thank the anonymous reviewers for the extended work which helped a lot to improve this contribution.

## REFERENCES

[1] D. R. Begault, “Challenges to the successful implementation of 3-D sound,” *Journal of the Audio Engineering Society*, vol. 39, no. 11, pp. 864–870, 1991.

[2] M. Naef, O. Staadt, and M. Gross, “Spatialized audio rendering for immersive virtual environments,” in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST ’02)*, pp. 65–72, Hong Kong, November 2002.

[3] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, “The CAVE: audio visual experience automatic virtual environment,” *Communications of the ACM*, vol. 35, no. 6, pp. 65–72, 1992.

[4] D. A. Burgess and J. C. Verlinden, “An architecture for spatial audio servers,” in *Proceedings of Virtual Reality Systems Conference (Fall ’93)*, New York, NY, USA, November 1993.

[5] J. D. Mulder and E. H. Dooijes, “Spatial audio in graphical applications,” in *Visualization in Scientific Computing*, M. Gobel, H. Muller, and B. Urban, Eds., pp. 215–229, Springer, Wien, Austria, 1994.

[6] Lake Huron, 2005, http://www.lake.com.au/.

[7] L. Savioja, *Modeling Techniques for Virtual Acoustics*, Ph.D. thesis, Helsinki University of Technology, Helsinki, Finland, December 1999.

[8] L. Savioja, J. Huopaniemi, T. Lokki, and R. Va ¨an ¨ anen, “Creating interactive virtual acoustic environments,” *Journal of the Audio Engineering Society*, vol. 47, no. 9, pp. 675–705, 1999.

[9] T. Funkhouser, P. Min, and I. Carlbom, “Real-time acoustic modeling for distributed virtual environments,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’99)*, pp. 365–374, Los Angeles, Calif, USA, August 1999.

[10] R. L. Storms, “Npsnet-3D Sound Server: An Effective Use of the Auditory Channel,” 1995.

[11] H. Kuttruff, *Room Acoustics*, Elsevier Science Publisher, New York, NY, USA, 4th edition, 2000.

[12] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[13] J. Borish, “Extension of the image model to arbitrary polyhedra,” *The Journal of the Acoustical Society of America*, vol. 75, no. 6, pp. 1827–1836, 1984.

[14] B.-I. L. Dalenback, “Room acoustic prediction based on a unified treatment of diffuse and specular reflection,” *The Journal of the Acoustical Society of America*, vol. 100, no. 2, pp. 899909, 1996.

[15] P.-A. Forsberg, “Fully discrete ray tracing,” *Applied Acoustics*,vol. 18, no. 6, pp. 393–397, 1985.

[16] T. Funkhouser, N. Tsingos, I. Carlbom, et al., “A beam tracing method for interactive architectural acoustics,” *The Journal of the Acoustical Society of America*, vol. 115, no. 2, pp. 739–756,2004.

[17] G. M. Naylor, “ODEON—another hybrid room acoustical model,” *Applied Acoustics*, vol. 38, no. 2–4, pp. 131–143, 1993.

[18] U. M. Stephenson, “Quantized pyramidal beam tracing a new algorithm for room acoustics and noise immission prognosis,” *Acta Acustica United with Acustica*, vol. 82, no. 3, pp. 517–525, 1996.

[19] D. van Maercke, “Simulation of sound fields in time and frequency domain using a geometrical model,” in *Proceedings of the 12th International Congress on Acoustics (ICA ’86)*, vol. 2, Toronto, Ontario, Canada, July 1986, paper E11-7.

[20] M. Vorlander, “Simulation of the transient and steady state sound propagation in rooms using a new combined sound particle—image source algorithm,” *The Journal of the Acoustical Society of America*, vol. 86, pp. 172–178, 1989.

[21] I. Bork, “A comparison of room simulation software—the 2nd round Robin on room acoustical computer simulation,” *Acta Acustica United with Acustica*, vol. 86, no. 6, pp. 943–956, 2000.

[22] M. Vorlander, “International round Robin on room acoustical computer simulations,” in *Proceedings of the 15th International Congress on Acoustics (ICA ’95)*, pp. 689–692, Trondheim, Norway, June 1995.

[23] H. Kuttruff, “A simple iteration scheme for the computation of decay constants in enclosures with diffusely reflecting boundaries,” *The Journal of the Acoustical Society of America*, vol. 98,no. 1, pp. 288–293, 1995.

[24] C. L. Christensen and J. H. Rindel, “A new scattering method that combines roughness and diffraction effects,” in *Forum Acousticum*, Budapest, Hungary, 2005.

[25] R. Heinz, “Binaural room simulation based on an image source model with addition of statistical methods to include the diffuse sound scattering of walls and to predict the reverberant tail,” *Applied Acoustics*, vol. 38, no. 2–4, pp. 145–159,1993.

[26] Y. W. Lam, “A comparison of three reflection modelling methods used in room acoustics computer models,” *The Journal of the Acoustical Society of America*, vol. 100, no. 4, pp. 2181–2192, 1996.

[27] M. Vorlander, “Ein Strahlverfolgungsverfahren zur Berech- nung von Schallfeldern in Raumen,” ¨ *Acustica*, vol. 65, no. 3,pp. 138–148, 1988.

[28] J. S. Suh and P. A. Nelson, “Measurement of transient responseof rooms and comparison with geometrical acoustic models,”*The Journal of the Acoustical Society of America*, vol. 105, no. 4, pp. 2304–2317, 1999.

[29] U. P. Svensson, R. I. Fred, and J. Vanderkooy, “An analytic secondary source model of edge diffraction impulse responses,” *The Journal of the Acoustical Society of America*, vol. 106, no. 5, pp. 2331–2344, 1999.

[30] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom, “Modeling acoustics in virtual environments using the uniform theory of diffraction,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’01)*, pp. 545–552, Los Angeles, Calif, USA, August 2001.

[31] U. M. Stephenson, *Beugungssimulation ohne Rechenzeitexplosion: die Methode der quantisierten Pyramidenstrahlen; ein neues Berechnungsverfahren fur Raumakustik und L armimmis- sionsprognose; Vergleiche, Ansatze, Losungen*, Ph.D. thesis, RWTH Aachen University, Aachen, Germany, 2004.

[32] M. Slater, A. Steed, and Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison Wesley, New York, NY, USA, 2001.

[33] L. Cremer and H. A. Muller, ¨ *Die wissenschaftlichen Grundlagen der Raumakustik—Band 1*, S. Hirzel, Stuttgart, Germany, 2nd edition, 1978.

[34] T. Akenine-Moller and E. Haines, ¨ *Real-Time Rendering*, A. K. Peters, Natick, Mass, USA, 2nd edition, 2002.

[35] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics, Principles and Practice*, Addison Wesley, Reading, Mass, USA, 2nd edition, 1996.

[36] R. Shumacker, R. Brand, M. Gilliland, and W. Sharp, “Study for applying computer-generated images to visual simulations,” Report AFHRL-TR-69-14, U.S. Air Force Human Resources Laboratory, San Antonio, Tex, USA, 1969.

[37] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2001.

[38] D. Schroder and T. Lentz, “Real-time processing of image sources using binary space partitioning,” *Journal of the Audio Engineering Society*, vol. 54, no. 7-8, pp. 604–619, 2006.

[39] R. Heinz, *Entwicklung und Beurteilung von computergestutzten Methoden zur binauralen Raumsimulation*, Ph.D. thesis, RWTH Aachen University, Aachen, Germany, 1994.

[40] J. S. Bradley and G. A. Soulodre, “The influence of late arriving energy on spatial impression,” *The Journal of the Acoustical Society of America*, vol. 97, no. 4, pp. 2263–2271, 1995.

[41] J. H. Rindel, “Evaluation of room acoustic qualities and defects by use of auralization,” in *Proceedings of the 148th*

*Meeting of the Acoustical Society of America*, San Diego, Calif, USA,November 2004.

[42] D. Schroder, P. Dross, and M. Vorlander, “A fast reverberation estimator for virtual environments,” in *Proceedings of the AES 30th International Conference*, Saariselka, Finland, March 2007.

[43] T. Brookes and C. Treble, “The effect of non-symmetrical left/right recording pinnae on the perceived externalisation of binaural recordings,” in *Proceedings of the 118th Audio Engineering Society Convention*, Barcelona, Spain, May 2005.

[44] D. S. Brungart, W. M. Rabinowitz, and N. I. Durlach, “Auditory localization of a nearby point source,” *The Journal of the Acoustical Society of America*, vol. 100, no. 4, p. 2593, 1996.

[45] A. Kulkarni and H. S. Colburn, “Role of spectral detail in sound-source localization,” *Nature*, vol. 396, no. 6713, pp. 747–749, 1998.

[46] H. Lehnert and M. Richter, “Auditory virtual environment: simplified treatment of reflections,” in *Proceedings of the 15th International Congress on Acoustics (ICA ’95)*, Trondheim,
Norway, June 1995.

[47] G. Romanenko and M. Vorlander, “Employment of spherical wave reflection coefficient in room acoustics,” in *IoA Symposium Surface Acoustics*, Salford, UK, 2003.

[48] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, “Surroundscreen projection-based virtual reality: the design and implementation of the CAVE,” in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’93)*, pp. 135–142, ACM Press, Anaheim, Calif, USA, August 1993.

[49] B. B. Bauer, “Stereophonic earphones and binaural loudspeakers,” *Journal of the Audio Engineering Society*, vol. 9, no. 2, pp. 148–151, 1961.

[50] O. Kirkeby, P. A. Nelson, and H. Hamada, “Local sound field reproduction using two closely spaced loudspeakers,” *The Journal of the Acoustical Society of America*, vol. 104, no. 4, pp. 1973–1981, 1998.

[51] H. Møller, “Reproduction of artificial-head recordings through loudspeakers,” *Journal of the Audio Engineering Society*, vol. 37, no. 1-2, pp. 30–33, 1989.

[52] W. G. Gardner, *3-D audio using loudspeakers*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1997.

[53] T. Lentz and O. Schmitz, “Realisation of an adaptive cross-talk cancellation system for a moving listener,” in *Proceedings of the 21st Audio Engineering Society Conference*, St. Petersburg, Russia, June 2002.

[54] T. Lentz and G. K. Behler, “Dynamic cross-talk cancellation for binaural synthesis in virtual reality environments,” in *Proceedings of the 117th Audio Engineering Society Convention*, San Francisco, Calif, USA, October 2004.

[55] Steinberg, “ASIO 2.0 Audio Streaming Input Output Development Kit,” 2004.

[56] T. Lentz, “Dynamic crosstalk cancellation for binaural synthesis in virtual reality environments,” *Journal of the Audio Engineering Society*, vol. 54, no. 4, pp. 283–294, 2006.

[57] T. Takeuchi, P. Nelson, O. Kirkeby, and H. Hamada, “The effects of reflections on the performance of virtual acoustic imaging systems,” in *Proceedings of the International Symposium on Active Control of Sound and Vibration (ACTIVE ’97)*, pp. 955–966, Budapest, Hungary, August 1997.

[58] D. B. Ward, “On the performance of acoustic crosstalk cancellation in a reverberant environment,” *The Journal of the Acoustical Society of America*, vol. 110, no. 2, pp. 1195–1198, 2001.

[59] T. Lentz, J. Sokoll, and I. Assenmacher, “Performance of spatial audio using dynamic cross-talk cancellation,” in *Proceedings of the 119th Audio Engineering Society Convention*, New York, NY, USA, October 2005.

[60] W. G. Gardner, “Efficient convolution without input-output delay,” *Journal of the Audio Engineering Society*, vol. 43, no. 3, pp. 127–136, 1995.

[61] J. J. La Viola Jr., “A testbed for studying and choosing predictive tracking algorithms in virtual environments,” in *Proceedings of the 7th International Immersive Projection Technologies Workshop, 9th Eurographics Workshop on Virtual Environments*, pp. 189–198, Zurich, Switzerland, May 2003.

[62] R. Azuma and G. Bishop, “A frequency-domain analysis of head-motion prediction,” in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’95)*, pp. 401–408, ACM Press, Los Angeles, Calif, USA, August 1995.

[63] L. Chai, W. A. Hoff, and T. Vincent, “Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality,” *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 5, pp. 474–492, 2002.

[64] J.-R. Wu and M. Ouhyoung, “A 3D tracking experiment on latency and its compensation methods in virtual environments,” in *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology (UIST ’95)*, pp. 41–49, ACM Press, Pittsburgh, Pa, USA, November 1995.

[65] I. B. Witew, “Spatial variation of lateral measures in different concert halls,” in *Proceedings of the 18th International Congress on Acoustics (ICA ’04)*, vol. 4, p. 2949, Kyoto, Japan, April 2004.

[66] R. Azuma and G. Bishop, “Improving static and dynamic registration in an optical see-through HMD,” in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’94)*, pp. 197–204, ACM Press, New York, NY, USA, July 1994.

[67] W. Pompetzki, *Psychoakustische Verifikation von Computermodellen zur binauralen Raumsimulation*, Ph.D. thesis, RuhrUniversitat Bochum, Bochum, Germany, 1993. ¨

[68] M. Vorlander and E. Mommertz, “Definition and measurement of random-incidence scattering coefficients,” *Applied Acoustics*, vol. 60, no. 2, pp. 187–199, 2000.

[69] ISO 354, “Acoustics, Measurement of sound absorption in a reverberant room,” 2003.

[70] ISO/DIS 17497-1, “Acoustics Measurement of the sound scattering properties of surfaces—part 1: measurement of the randomincidence scattering coefficient in a reverberation room”.

[71] N. Tsingos, “Scalable perceptual mixing and filtering of audio signals using an augmented spectral representation,” in *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx ’05)*, Madrid, Spain, September 2005.

[72] N. Tsingos, E. Gallo, and G. Drettakis, “Perceptual audio rendering of complex virtual environments,” in *Proceedings of the 31st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’04)*, pp. 249–258, Los Angeles, Calif, USA, August 2004.

# 附录 B 外文翻译译文

## 集成声场虚拟现实系统模拟和再现

实时音频渲染系统介绍它结合了全特定房间模拟、动态串音消除，并为虚拟声学成像多轨立体合成。该系统适用于任何房间形状（正常、长、平、耦合），独立于扩散声场的先验假设。这提供了模拟的室内或室外空间分布，可自由移动的来源和在虚拟环境中移动收听者的可能性。除此之外，近源可以通过使用测得的近场的 HRTF 来模拟。再现组件由动态串音消除无耳机再现。该项目的重点主要是所有参与的子系统的集成和交互。它表明，该系统能够实时房间仿真和繁殖，因此，可以用作用于虚拟现实应用中的进一步研究了可靠的平台。

## 1 引言

虚拟现实（VR）是在与该用户可以操作并实时交互的计算机生成的环境。 VR 的一个特征是一台计算机和人类之间的三维和多模式接口。在科学，工程和娱乐领域，这些工具是源远流长的在几个应用程序。可视化 VR 通常主要关注的技术。在 VR（可听化，声波处理）声学不在场相同的程度，并且通常只是加入作为一种效果，而没有任何合理的参考虚拟场景。具有实时性能可听化的方法可以集成到的技术“虚拟现实”。

产生用于各个感官线索（3D 图像、3D 音频等）的过程被称为“呈现”。显然，相互作用的简单的场景，例如，当一个人离开房间和关闭的门，需要复杂的室内声学和隔音的车型。否则，它是可能的着色，响度，以及内部和室之间的声音的音色没有充分表示。另一个例子是后面的阻挡或结构的一个开口内的探测对象的交互移动，从而使该对象不再是可见的，但是可以是通过衍射听到。

### 1.1 声场建模

产生逼真的声学感知，定位和识别的任务，是一个很大的挑战。与此相反的视觉表示，声学处理涉及三个数量级（20 赫兹到 20 千赫和波长从约 20μm 至 2 厘米）的频率范围。小波长也不大波长近似也不可以假设与一般的有效性。不同物理法则，也就是衍射在低频，在高频散射和镜面反射已被应用到生成一个基于物理的声场模拟。因此，从物理角度（这意味着，更不用说实现的挑战），建模与仿真的问题

确切的虚拟声音的是由数量　　级比创建的视觉图像的任务更加困难。这可能是在延迟执行在虚拟环境中的声学部件的原因。

目前，个人电脑只是能够模拟实时似是而非的声学效果。为了达到这一目标，许多近似仍然必须作出。对于产生的声音的终极目的不是在物理上完全正确的，但是感知可信的。关于人的声音感觉知识，因此，对于评估声音一个非常重要的先决条件。

环境本身，外部事件，和非常自己的行动很重要，反馈的认知是通过听证活动的支持。特别是在虚拟现实环境中，用户的浸没到计算机生成的景色是一个非常重要的方面。在这个意义上，浸渍可以定义为寻址以自然的方式，所有人的感官子系统。作为收件人，

人类评估总的声音的不同特点分成单个对象。此外，他们评估环境本身，它的大小和平均吸附（嵌合状态）。在声学场景在一个房间里，这可能是典型的为广大的虚拟现实应用的情况下，所有的这些主观印象的身体足够的代表性，因此必须进行模拟，回归和转载。合理性可以，但是，只能为特定的环境中定义。因此，声场建模的一般方法需要在广泛的房间，建筑物或室外环境的物理基础和适用性。

## 1.2 再生产

听觉组件附加地强制用户的身临其境的感受由于环境的通过的空间表示的理解。除了声场建模本身的信号的一个适当的重放是非常重要的。的目标是运输中包含的信号中的所有空间提示在听觉正确的方式给听众的耳朵。如上所述，着色，音量和音色是必不可少的，而且声音的方向和它的反射都需要一个至少似是而非场景表达。在空间信号的定向信息是

非常重要的，以表示其全部复杂性的余地。此外，这是通过动态适于双耳演示其使得听者移动和转动产生的虚拟世界中的支持。

## 1.3 系统

在这方面的贡献，我们描述了声场的建模和安装在亚琛工业大学的虚拟现实系统的3D 声音再现的物理算法方法（见图 1）。该系统是在第一版本中实现。它是开放的实时任何扩展的物理声场建模，并且是独立于任何特定的视觉虚拟现实显示技术，例如，洞穴般的显示器或基于桌面的解决方案。名为 VirKopf 我们的 3D 音　　响系统曾在亚琛工业大学声学技术协会（ITA），已经付诸实施，作为分布式架构。对于任何一个房间声学仿真，VirKopf 使用软件 RAVEN（室内声学为虚拟环境）作为网络服务（见第 2.1 节）。显然，视频和音频处理占用大量计算资源的各个子系统，并按照今天的标准，那是不现实做一台机器上所有的处理。出于这个原因，音频系统实现的视频和音频数据的上由一个网络互连的专用机的计算。这种想法是显而易见的，已经成功地通过实施。甚至有可商购的溶液，其已经通过专用硬件实现，可以经由网络接口　　一起使用，例如，休伦湖机。由一个网络接口绑定声学再现分量的其他实例可以在与 DIVA 项目或芬克豪泽的光束连接中找到跟踪。其他的方法，还没有被实现为网络客户端 - 服务器架构，但依靠　　特殊的硬件设置。

该 VirKopf 系统不同于在某些方面这些方法。一个主要的区别是 VirKopf 系统的重点，提供对移动听众双耳的声音体验的可能性，而无需任何身临其境的虚拟现实环境中的耳机。其次，它不上的任何约束的硬件要求顶端实现，诸如特定的 DSP 技术存在用于音频处理。该 VirKopf 系统实现纯软件的方式，并可以关闭的，现成的定制 PC 硬件中使用。除此之外，该系统不依赖于专门放置扩音器或大量扩音器。四个扬声器足以使用双耳方法单个用户创建一个周围的声学虚拟环境。

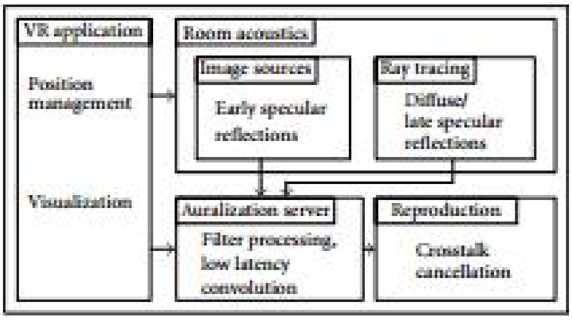


图 1：系统组件

## 2 室内声学模拟

由于多种原因，这是不能在这里所有的细节进行说明，几何音响效果是用于在室内声学可听化的最重要的模式。波模式会更精确，但只有几何声学的近似值和相应的算法提供一个机会，以模拟在实时应用室内脉冲响应。按照这种解释，延迟线模型，辐射或其他被认为是基本几何，以及因为波的传播降低到从墙壁能源过渡的时域方法墙。在几何声学，确定性和随机性方法可用。今天所使用的所有确定性模拟模型基于图像源的物理模型。它们在声音路径如何通过使用前向（射线）跟踪或扭转结构确定的方式不同。这种类型的算法的变体

是混合光线追踪，追踪梁，金字塔跟踪，等等。从图像状模型的脉冲响应由相应布置成他们的延迟和幅度过滤狄拉克脉冲并且以一定的时间分辨率进行采样。在模拟程序比对，很快就清楚，由于室内声学地面和障碍物的一个非常重要的方面纯净的图像源模型会产生过于粗糙的房间物理声场近似散射忽略不计。

可以看出的是，从为了两个或三个反射，散射，甚至在配有相当光滑的表面变得在房间脉冲响应的时间发展的显着效果（见图 2）。幸运的是，散射声音的特定方向分布是在第二或第三反射顺序之后无关并能很好地被假定为朗伯散射。但是，在配有高吸收的物质如录音室，其中定向扩散系数有关的特殊情况下，不同的散射模型必须被使用。表面散射问题的解决方案是由要么随机光线跟踪或辐射中给出。此外，事实上，图像源是一个良好的近似为完全反射或吸收少的表面常常被遗忘。图像的近似，但是，在大房间有效期至少为源，墙壁，和接收器之间的大的距离。波的另一个作用物理的衍射可以引入几何声学，但到目前为止，在线仿真一直被限制在固定声源。出现的主要问题，然而，延长衍射模式，以更高的订单时。相距

从室外应用，衍射尚未在应用，如室内声学的情况下实现的。它应，然而，提及的是，许多算法细节已经发表在声场渲染领域迄今。新算法方案，例如那些由提出尚未实现。这里应当记住的是，这两个基本的物理方法确定性声图像和随机散射应当考虑到的声场模型的现实物理行为一定的性能。声音传送以及衍射必须在何处声音通过孔透射耦合室，在走廊，或例来实现。

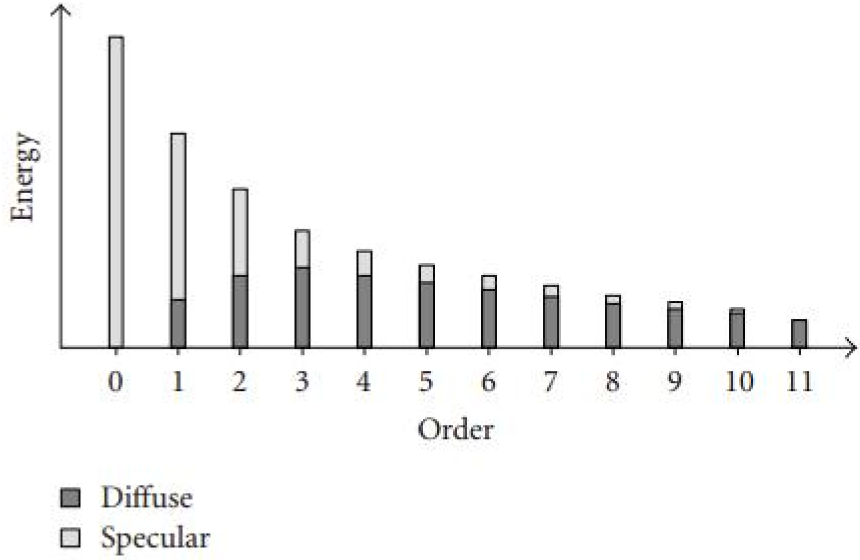


图 2：镜面转化为漫反射的声能，通过一个例子来说明

## 2.1 实时执行能力

任何房间声学仿真时，应考虑到声音的上述物理方面的房间。典型地，软件可用于计算的静态源和在几秒钟或几分钟内听众位置的室内脉冲响应。但是，无限制的运动接收器和几何和物理边界内的声源的是任何交互式在线可听化基本要求。此外，在风景，例如任何相互作用，开门到一个相邻的房间，房间的变化的在线更新'

模态结构应通过模拟来提供，以产生虚拟世界[32]的具有高可信度。

目前，被称为 RAVEN 房间声学仿真软件正在以我院研制。该软件的目的是满足上述所有条件的听觉部件的逼真的模拟，但是，对于实时能力。特别提供了实现的实时房间声学模拟的可能性将在下面的章节进行说明。 RAVEN 基本上是由 Vorlander 升级和增强混合室声学模拟方法[20]，将其进一步通过亨氏[25]延伸。处理室（参见 2.2 节）任意数量的一个非常灵活和快速进入子菜单框架已被纳入以获得高互动的仿真水平，以达到在一定的约束条件算法实时性（参见 5.2 节）。图片来源用于为了确定早期反射（参见 2.3 节）提供的主要声源的模拟过程中最准确的定位（优先效应[33]）。散射和混响是通过一种改进的随机射线追踪法的手段，这将在第 2.4 节进一步描述估计上线

## 2.2 场景分割

房间的测定“的声音反射需要光线和房间之间的相交测试数量巨大”几何几何以来声学方法把声波为“光”射线。适用于实时这些方法中，都需要一个有效的表示和声音射线和房间几何之间的空间关系的测定数据结构。

这些数据结构分层组织几何某些n维空间，通常递归性质的显着地加速操作的查询，如拣出算法，相交测试，或碰撞检测。我们的可听化框架包含了一个预处理阶段，其通过使用二进制空间分割将每个单间的几何形状为一个灵活的数据结构（BSP）的模拟过程中快速相交测试的树木。此外，场景图，这基本上是在单间的数据结构的顶部上的逻

辑层的概念，是用于使该框架适用于房间的任意数量和获得的房间声学模拟高水平的交互性。

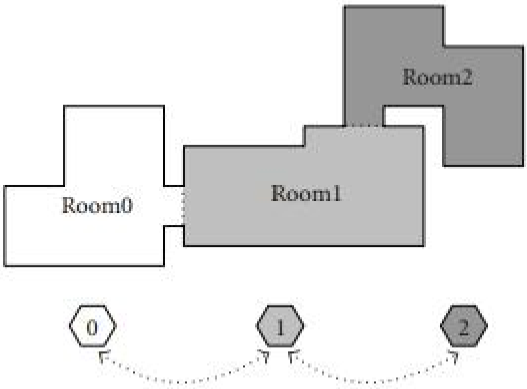


图 3：风景被分成三个房间，这是由场景图的节点代表（通过六边形表示）。房间由 2 个门户网站（room0/ ROOM1 和 ROOM1/室 2 连接到他们的隔壁房间通过虚线表示线）。

## 2.2.1 场景图结构

实现了高效的数据处理的房间的任意数目，场景图的概念已被使用。的场景图是根据房间的邻接链接节点的集合。

节点包含相应的子场景的逻辑和空间表示。每个节点是由所谓的门户，其中代表连接各个房间的实体链接到它的邻居，例如门或窗（见图 3）。应当指出的是门户的单个节点的数量没有限制，因此，景色可以非常灵活地划分成子场景。使用门户网站的巨大优势是两个国家可能发生的二进制性质。国家“积极的”连接门户网站定义的两个节点，而关闭特定链路的状态“被动”的削减。这提供了交互性的高水平的室内声学模拟作为房间的街区可上线被改变，例如，门可被打开或关闭。另外，关于门户状态的信息可被利用通过忽略间客房都声学不感兴趣，例如，客房，是出界对当前接收器的位置，加快上线室内声场模拟过程中所需的任何测试。

## 2.3 图片来源方法

传统的图像源（IS）方法提供了一种非常灵活的数据结构，因为，举例来说，一次的声源，其相应的图像源的在线运动被支撑并且可以在几毫秒内被更新的概念。不幸的是，该方法将失败作为计算费用由图像源与指数增长为主，以模拟大风景增加客房数量，也就是说，多边形和反思秩序。施加了 IS 方法进行处理的房间将导致在 IS 爆炸的任意数量，这会使得一个大的虚拟 ENVIRON-彪实时约束内不可能的模拟由于极端数 IS 进行在线测试在可听性。

然而，场景图数据结构（见第 2.2.1 节）提供了根据由排序整个集的当前门户配置的预先计算潜在可听的子集的可能性取决于他们从起源的室。这可以很容易地通过预处

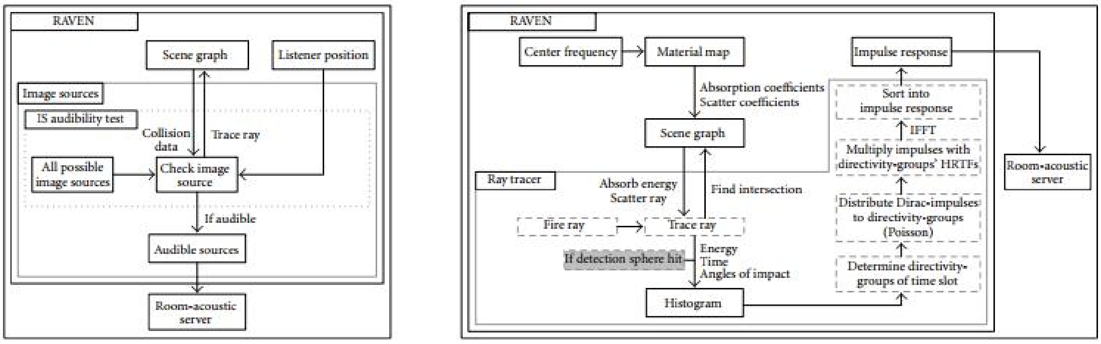
理设置场景 S，其中 S 是一组 n 个室的功率来完成。 S 的功率设定包含 2n 个元素，并且每一子集，即，家庭组 S 是指 n 位数字，其中的第 m 位是指活性或 S 的第 m 个房间的不活动之后，所有的 IS 通过收集信息被分类到相应的家庭组的 S 关于飞机的房间的 ID，他们已经反映上。图 5 示出示例性地含有三个室 R2，R1，R0 一个风景 S 的幂集 P 和 IS 的联子集，即，P（S）= {{主源}，{IS（R 0）}， {IS（R 1）}，{IS（R1，R2）}，{IS（R 2）}，{IS（R 2，R 0）}，{IS（R 2，R 1）}，{IS（R2，R1，R0） }}。

在上线可听化，深度优先搜索场景图的确定的当前接收机的位置到达房间的 ID。这不包括这两种客房，是出由门户封锁边界和房间。这组室 ID 的由功率集合 P 编码来设置无效可达室，它们是声学不感兴趣。如果在本实施例的房间的情况下，R 2 为当前接收机的位置变得不可达，例如，有人关上门，仅是 P 的家庭集具有对可听化不包含的房间 ID R2 至被处理。作为其结果，由家庭的数目设置要对可听度测试滴从八至四个，即，P（0），P（1），P（2），P（3），这显然会导致一显著减少的计算时间。

期间模拟它必须被检查每个可能的可听图象源，这是如上所述判断是否是可听当前接收机的位置（参照图 4 的（a））。服用场景图的底层 BSP 树结构的很大的优点和高效率的树遍历策略，所需要的 IS 可听度测试可以做得非常 快（性能问题进行了更详细的第 5.2.1 节中讨论）。如果一个图像源上可听度测试当前接收机的位置，被要求的所有数据滤波计算（位置，交点，并击中材料）将存储在超级配合的容器“可听源”（参见图 4 的（a ））。

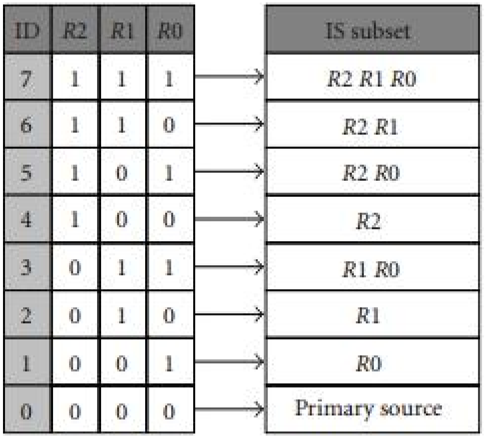
## 2.4 光线追踪

扩散声场的计算是基于由亨氏提出的随机光线跟踪算法。构建从光线追踪数据双耳脉冲响应，亨氏假设混响是理想的弥漫。这种假设，但是，太 粗糙，如果房间的几何是极长的或平的，如果它包含象列或隐私屏幕对象。房间的声学缺陷，如（扑）回声将不被发现[40，41]。为更现实的房间的声学模拟，该算法已经在某种程度上改变，使得这些效果都考虑在内（参见图 4 的（b））。



(a) Image sources

(b) Ray tracing



| ID | R2 | R1 | R0 | IS subset |
|---|---|---|---|---|
| 7 | 1 | 1 | 1 | R2 R1 R0 |
| 6 | 1 | 1 | 0 | R2 R1 |
| 5 | 1 | 0 | 1 | R2 R0 |
| 4 | 1 | 0 | 0 | R2 |
| 3 | 0 | 1 | 1 | R1 R0 |
| 2 | 0 | 1 | 0 | R1 |
| 1 | 0 | 0 | 1 | R0 |
| 0 | 0 | 0 | 0 | Primary source |

图 4:（1）图像源可听度测试，散射和混响（b）的估计

图 5：是一个三房的情况/室组合功率设定 P（S）。所有的 IS 分类成取决于它们已被从所生成房间组合包封的容器中。这一方面是在实时虚拟声学的创新，这是被为是一个重要的扩展感知维度。

基于 BSP 射线跟踪模拟通过从以随机的角度各声源，其中每个粒子携带的能量源方向性依赖量发光颗粒的有限数目开始。每个粒子失去能量，而内部的房间传播因空气的吸收和上壁发生反射，或者镜面或漫反射，和其它几何对象，即，声音的材料依赖性吸收。粒子被尽快粒子的能量被下一个预定的阈值降低终止。一个时刻 $t_0$，它表示图象源的截止时间之前，仅检测到颗粒，其已经反映了漫历史镜面，以保持正确的能量平衡。 $t_0$ 之后，反射类型的所有可能的排列被处理（例如，漫反射，镜面反射，漫射，漫射等）。

光线跟踪的每个频带由于依赖于频率的吸收和散射系数，这导致所谓直方图三维数据容器进行。该直方图被认为是充满活力的空间脉冲响应的时间包络。一直方图的单个字段包含有关射线的信息（在抵达时，时间精力和影响的角度）的时间间隔Δt 的离散频率间隔 FB 中达到了检测范围。首先，对于具有不同的频率领域的平均能量但在相同的时间间隔被计算，以获得短时能量谱密度。此步骤也被用来创建用于相应光线随时间的光线指向性分布：对于每个时隙，检测球被分成均匀分布的分区，所谓指向性基团。如果射线击中球时，光线的上冲击剩余能量被添加到根据其时间和到达方向的相应球体的方向性基（参见图 6）。

此能量分布被用来确定每个方向性组和每个时间间隔Δt 射线概率。然后将创建具有等于反射为给定的房间和给定时间间隔的速率的速率泊松过程。该方法的每个脉冲被分配给各个根据所确定的射线概率分布方向性基。在最后的步骤中，这是由一个泊松脉冲簇击中每个方向性基团被其相应 HRTF 乘以叠加到双耳信号，以及由能谱密度的平方根加权。在此之后，该信号被转换成时域。这是对直方图的每个时间步长完成，放在一起，完整的双耳脉冲响应。光线跟踪算法由室内声学服务器管理的是提供一种动态更新深度的可能性，用于确定漫声场部分（见第 3 节）。由于这方面的贡献侧重于整个系统的实现和性能，没有进一步的细节在这里呈现。快速执行和测试结果的详细描述中可以找到。

# 3 滤波处理

对于其中侦听允许移动，转动，和相互作用与所呈现的风景并且其中源也可以被移动，所述室内脉冲响应必须被快速更新动态可听化。这也成为结合全等视频图像更重要。因此，该滤波处理是实时过程的关键组成部分。整个过滤器结构被分成两部分。双耳房间脉冲响应中最重要的部分是含有直接声音和房间的早期反射的第一部分。这些早期反射是由所计算出的图像源所表示，并以在其具有到足以双耳处理的速率被更新。出于这个原因，在室内声学服务器和可听化服务器之间的操作界面的当前可听源的列表。房间脉冲响应的第二部分所计算的室内声学服务器（或簇）上，以减少所需要的时间通过网络传输，因为计算出室内脉冲响应所需要的数据量小于所得到的过滤器本身显著更高。

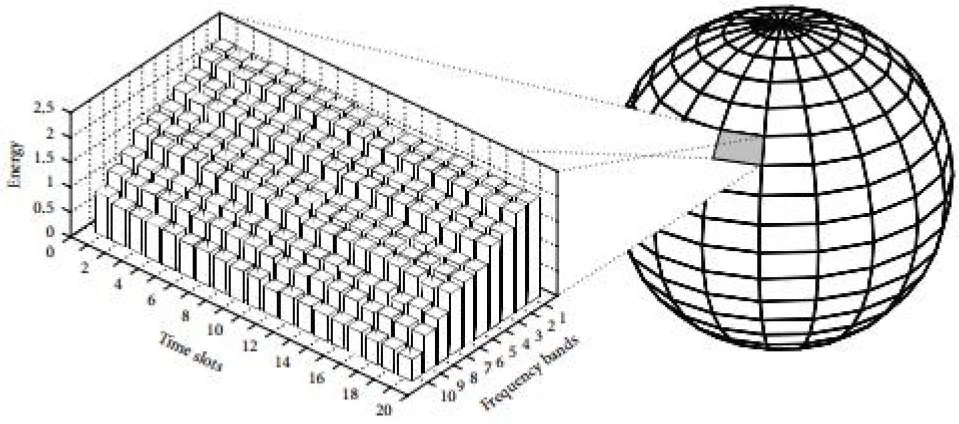


图 6：单方向性组的直方图例

## 3.1 图片来源

完整的脉冲响应的每一个部分，无论是直接的声音或由一个或多个壁反射的声音，通过几个过滤器元件运行，如图 7 中的对数频率的元素，如方向性，墙壁，和空气的吸收是过滤器表示与第三倍频带刻度与 31 的值从 20 赫兹到 20 千赫。这些过滤器包含使得仅需要一个单一的乘法没有相位信息。使用对数表示的缺点是内插的必要性与 HRTF 乘以所得滤波器。但这仍然不如使用线性表示所有元素，尤其是如果有更多的壁式过滤器具有被考虑的特定反射作为计算昂贵。

到目前为止，墙壁吸收过滤器是独立的声音入射角度，这是室内声学模型的一个共同的假设。它可以扩展，如果必要考虑角度依赖性的数据。通过使用图像源模型计算反射将由因子来衰减的这是由漫反射分布的能量。在漫反射将由光线跟踪算法来处理，（参见 3.2 节）。

在室内空间，尤其是宽敞的大厅，声音的另一个重要影响是源的方向性。这是即使对于这里不仅听者可以移动并与风景但是其中源也可以移动或转动相互作用的动态可听化更重要。整个产生的声音场景的自然被每个动力方面改善被考虑。程序接受任何空间分辨率的外部指向数据库，内部数据库具有 5 度方位角和仰角的空间分辨率。该数据库包含一个歌手和多种天然乐器的方向性。此外，它可以手动生成一个方向性。空气吸收滤波器仅有距离有关，并且还适用于直接的声音，这是对于收听和源极之间的距离远必不可少。

在每一个滤波器的通，这表示，到现在为止的最后，一个单声道信号，的 HRTF 已

被用于产生包含所有方向信息的双耳头相关信号。由 VirKopf 系统中使用的所有的 HRTF 与 ITA 为全球的人工头进行测量，由于不对称羽片和头部的几何形状。不对称羽片导致对所生成的虚拟源的感知外在化起到积极作用。一股强烈的冲动成分，如直达声带有一间客房源的最重要的空间信息。为了提供更好的分辨率，即使在低的频率，更高的分辨率的 HRTF 被用于直接声音。 FIR 滤波器长度选择为 512 水龙头。由于这样的事实，该滤波处理是在频域中进行，所述过滤器由对应于 86 赫兹的线性分辨率 257 复频域值表示。

此外，该数据库不仅包含在一个特定的距离，但也近场的 HRTF 测量的 HRTF。这提供了一种自然的方式模拟近到头部来源的可能性。试验表明，在一定距离的增加耳间水平差（ILD）变为可听为 1.5μm 或更接近的头部。该试验中的 ITA 的半消声室中进行，检查其中不同的近场的 HRTF 必须应用的范围。听众提出与那些从相应测量的 HRTF 在两个标准，即，源的感知位置和信号的任何着色比较从模拟化 HRTF 的信号。从远场的 HRTF（在两米的距离测量）制备模拟的 HRTF 用一个简单的水平修正同样适用于两个通道。所有九个听众报告了关于横向声中发病率正在距离小于 1.5 米接近的情况下的差异。对于正面的声音的发生率没有差异报告的距离比 0.6 米接近的情况下。这些结果非常类似于在其他实验室中进行的研究得到的结果，例如，[44]。因此，在 0.2 微米的距离，0.3M，0.4M，0.5M，0.75 男，1.0M，1.5 米到 2.0 米，测量的 HRTF。数据库的空间分辨率为方位角 1 度，5 度为两个直接声音和反射的仰角。

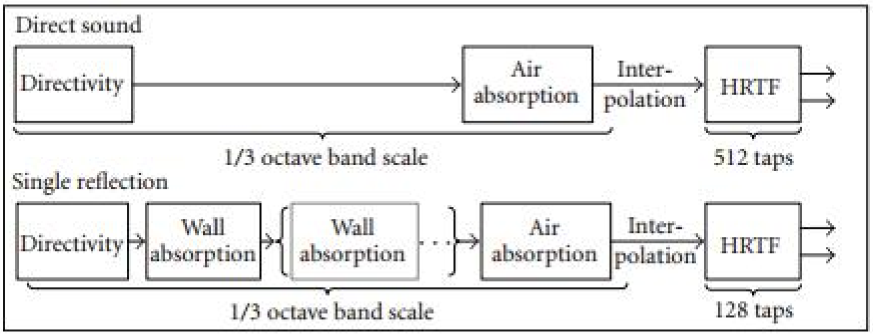


Figure 7: Filter elements for direct sound and reflections.

用于图像源的贡献 128 抽头的 FIR 滤波器长度比为直达声低，但仍高于在文献中找到的限制。可以在中找到关于本地化降低滤波器长度的影响的调查。作为直接声音，所述滤波处理是与 65 个复数值的相应滤波器表示在频域进行。

使用 128 FIR 系数导致同一定位结果，但带来了相当大的减少的处理时间（见表 3）。此进行了测试，以及在内部收听体验，但是也全等其他实验室的发现，即。图像的空间表示源是通过使用在 2.0 米测量的 HRTF 来实现。在这种情况下，这并不意味着任何的简化，因为使用图像源的房间声学模拟不无论如何在到墙壁近距离（几个波长）有效。有关该主题的更详细的调查中可以找到。

## 3.2 光线追踪

如上所述，光线跟踪处理的双耳脉冲响应的计算上的射线，以减少其具有经由网络要传输的数据量的跟踪服务器完成。为了保持过滤器根据过滤嘴段，这是关系到时间对准的重要性上最新的可听化过程可以发送中断命令到模拟服务器。如果源或收听者的速

度太快，以完成适当的时隙内的过滤器的计算，运行的光线追踪处理将被停止。这意味着，滤光器的更新深度取决于听众或来源的运动。为了实现可中断的光线追踪处理中，有必要将整个滤波器长度分成几个部分。当光线到达指定时间标记，必要的数据，重新启动在该位置的射线将被保存，并计算下一射线。在完成所有的光线的计算之后，该过滤器将被处理到光线跟踪更新直方图中的信息的时间（这也可以是平行的过程中，如果由硬件提供）。在这个时候，也有可能向第一更新滤波器部分发送到可听化服务器，这意味着它可以完整的光线跟踪完成之前采取改变脉冲响应的前面部分考虑。在这一点上，光线跟踪过程将在中断标志决定的计算是否在过滤器的开始或在最后时间标记重新启动。对于头，光线追踪过程或来源的轻微或动作缓慢有足够的时间通过一个包含所有过滤时间段的完整的计算周期来运行。这也导致了仿真的准确度的水平的持续时间上升的事实听者站在大致相同的位置和源不移动。

## 4 再生系统

本文描述了房间声学建模的主要生殖系统是安装在洞穴般的环境，这是一个矩形，安装在亚琛工业大学的五面投影系统中设置。特殊形状使得能够通过在墙壁上的液晶投影机的 1200 像素和地板以及 360 度的水平视图的使用 1600 的全分辨率的。凸起体积的尺寸是 3.60×2.70×2.70 立方米得到 26.24 平方米，总投影屏幕面积。此外，通过圆极化使用被动立体允许轻便的眼镜。头和交互设备跟踪是由一个光学跟踪系统来实现。该显示系统的设置是一种改进的实现，是与针对性开发，以提高用户的认可最大限度地减少附件及产权负担的制度。在这个意义上，多，洞穴般的环境在近几年获得了信誉已被归因于一个事实，他们试图绝对非侵入式虚拟现实系统。其结果是，一个基于扬声器的声学再现系统似乎是在 CAVE 状环境声学成像最期望的解决方案。用户应该能够步入虚拟场景没有太多的准备或校准，但仍沉浸在一个可信的环境。出于这个原因，上面描绘我们的洞穴般的环境与使用扩音器双耳再现系统扩展。

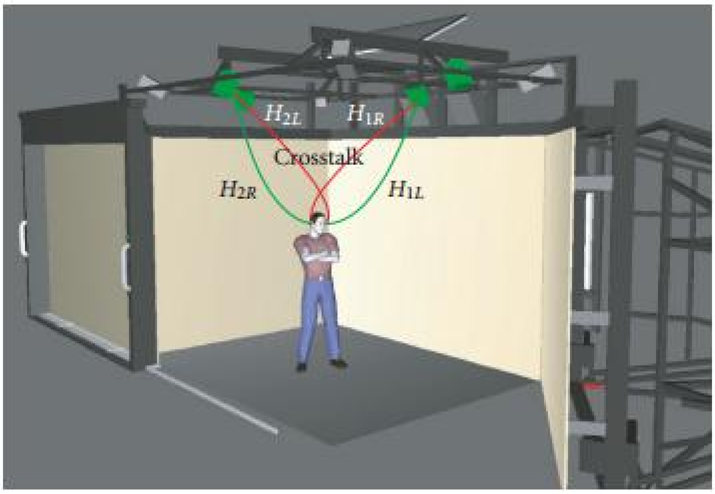


图 8：在亚琛工业大学的洞穴般的环境。四个扬声器被安装在该系统的顶部机架。门，左边所示，和一个可移动的墙，如右图所示，可以关闭，以允许与没有屋顶的投影 360 度的视角。

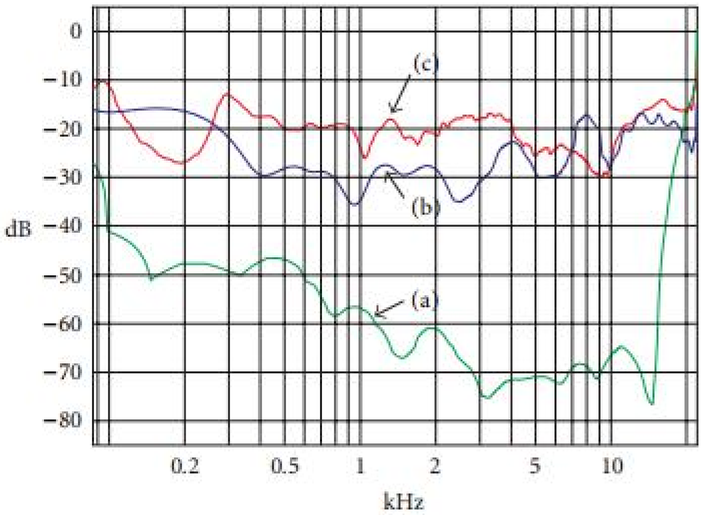


图 9：在使用的 1024 个抽头的滤波器长度可访问信道分离的测量。（a）=计算的，（b）=静态溶液，（c）=动态系统。

## 4.1 虚拟耳机

要重现在具有足够的信道分离的耳朵双耳信号而无需使用耳机，串音消除（CTC）系统时，需要。这样的环境下，用户应该能够四处走动，并把他的头反恐委员会的工作需要一个动态的 CTC 系统，该系统能够在听者的运动适应。动态的解决方案覆盖正常的静态串扰消除的甜蜜点的限制。图 8 显示了从扬声器到听者（H1L =传递函数扬声器 1 至左耳）的耳朵四种传输路径。正确的双耳的再现是指从左边输入到左耳的完整传输函数（参考点是耳道的入口处），包括传递函数 H1L 意指成为平坦的频谱。相同的适用于右传递路径，因此。由 H1R 和 H2L 指示的串扰已经由系统被取消。

由于虚拟环境的用户已经被跟踪，以产生正确的立体视觉影像，也能够在线计算四氯化碳滤波器对于用户的当前位置和方向。在运行时计算增强关于有效性区域并且几乎不能与预处理滤波器实现扬声器的设置的灵活性的 VirKopf 系统的灵活性。因此，需要包含“所有”可能的 HRTF 数据库。该 VirKopf 系统使用的数据库具有一个程度的空间分辨率为方位角（φ）和仰角（θ）。所述一化 HRTF 在 100 赫兹-20 千赫的频率范围内测量，从而允许在相同的频率范围的取消。应当提到的是取消在较高频率下更容易出错的扬声器的不对准并且还耳廓的个体差异。这也显示在图 9 中所示的曲线（c）该扬声器与头部之间的距离的影响的时间延迟和信号的电平。使用具有在一定的距离处测量的 HRTF 数据库，这两个参数必须被调整

通过修改滤波器的群延迟，并根据球面波衰减的实际距离的水平。

向用户提供一个完整的头部的旋转，两扬声器的设置将不会被足够的动态　　取消将仅由扬声器所跨越的角之间的工作英寸因此，双四氯化碳算法与 fourspeaker 安装程序已被开发出来，这是在[54]进一步的描述。四个扬声器，一个正常的双通道 CTC 系统八个组合是可能的，并且可以为听者的每个方向实现一个适当的取消。的角度依赖衰落是用来改变的有源扬声器两种配置的有效性重叠区域之间。

每个头跟踪信息是在系统更新时，头部的位置和方向的偏差相比这引起前述滤波器变化的计算给出的信息。自由的每度加权有自己的因素，然后总结出来的。

因此，阈值可以在六个自由度，位置值（ΔX，ΔY，ΔZ）进行参数化，和旋转的

值（Δφ，Δθ，Δρ）。当加权和大于 1 的横向运动，并在水平面头部转动最关键的，ΔX=ΔY 埃= 1cm 和Δφ= 1.0 度被选择主导滤波器更新的滤波器更新将被执行。阈值通常是指价值其中，超出上限的最后一次。所得迟滞防止两个滤波器之间的永久切换当一个固定的间隔两个过滤器和跟踪数据抖动之间确定的边界稍微它可能会发生。

一项所述的声音输出装置的基本要求是，信道工作绝对同步。否则，将计算出串扰路径不适合用给定的条件。此帐户，特殊音频协议 ASIO 由 Steinberg 专业音频记录的设计被选择为解决输出装置。

于理论上可以由动态系统所能达到的性能进行分类，静态系统的进行了测量，以具有用于实现信道分离现实参考。下绝对理想的情况下，一化 HRTF 用于计算串音消除滤波器是相同的再现期间（听众的个人的 HRTF）。在第一测试中，串音消除滤波器进行了用人工头的 HRTF 在固定位置进行处理。窗口化到一定的滤波器长度和平滑产生的信道间隔的限制。内部滤波计算长度被选择为 2048 个抽头，以便考虑到所引起到扬声器的距离的时间偏移。所述一化 HRTF 用 1/6 倍频程的带宽，以减少小倾角这可能导致通过反向滤波器问题平滑。在计算后，将过滤器组被截断为 1024 抽头，该动力系统的工作原理与相同长度的最终滤波器长度。然而，单过滤器之间的时间对准不受截断。使用此（截断）滤波器组和所述平滑化 HRTF 为基准计算出的信道分离在图 9 的曲线（a）的作图。此后，实现信道分离，在人造头部的耳朵，这尚未因为 HRTF 测量移动测量（图 9 曲线（b））。

在比较理想的参考的情况下，图 9 曲线（c）示出动态 CTC 系统的实现的信道分离。静态和动态系统之间的主要区别是一组用于滤波计算的 HRTF 的。动态系统具有选择从数据库相应 HRTF，并具有调节延迟，并根据该位置数据的水平。所有这些调整直接在这一点上测量的理想 HRTF 引起轻微的偏差。出于这个原因，动态系统的信道间隔是不一样高可通过用直接 HRTF 测量的系统来实现的之一。

串音消除的理论是基于再现在消声室环境的假设。然而，洞穴般的环境中投影壁由这降低了 CTC 系统的性能造成固体材料的反射。与我们的系统听力测试表明，主观定位性能还是非常不错的。另外，其它的实验室测试和不同的 CTC 系统显示更好的主观方面表现会比从测量可以预期的。一方面验证这种现象是通过该声源定位主要是由第一个到达的波前确定的优先效应;另一个方面是头部运动这给用户批准入射的感知方向的能力。更对我们的双耳演示和再现系统的性能的详细调查中可以找到。

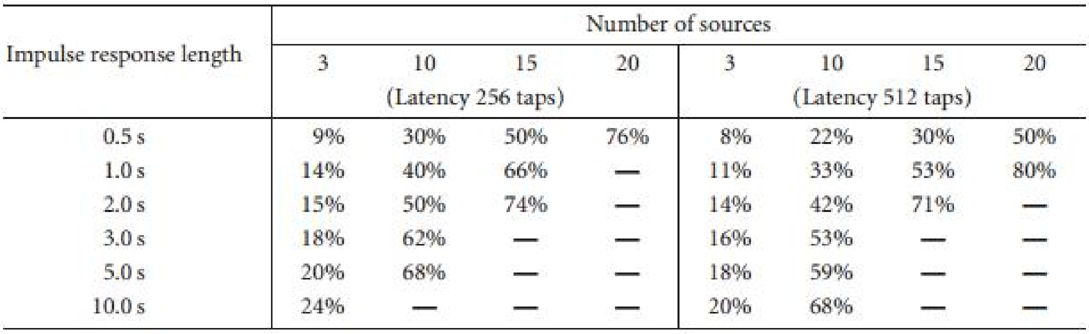
音频再现系统的等待时间是一个新的位置的更新和听者的取向，并且在其中所述输出信号与重新计算的过滤器所产生的时间点之间经过的时间。卷积的输出块长度（重叠保存）是 256 抽头，以及作为声音输出装置的所选择的缓冲器长度，导致两个缓冲开关之间的时间 5.8 毫秒以 44.1kHz 采样速率为单个块的呈现。新 CTC 过滤器组（1024 抽头）的计算需要我们的测试系统上 3.5 毫秒。在最坏的情况下，滤波器运算刚结束后的声音输出装置取出的下一个块，所以它花费的时间播放该块，直到更新的滤波器变为在输出端有效。这会导致一个块的等待时间。在这样的情况下，总的延迟时间累积到 9.3 毫秒。

## 4.2 低延迟卷积

需要的处理能力的高量的完整动态可听化系统的一部分是所述音频信号的卷积。纯 FIR 滤波不会引起额外的延迟除外过滤器的第一个冲动的延迟，但它也造成处理能力的

最高金额。超过 100 000 抽头或多个脉冲响应不能被实时处理的使用 FIR 滤波器在时域中的 PC 系统上。块卷积是降低了计算成本降至最低，但成比例地滤波器长度的延迟增加的方法。尽量减少卷积的等待时间的唯一方法是在完全脉冲响应的特殊调理过滤块。基本上，我们使用的在频域与在过滤器的开始小的块大小和增加的尺寸的过滤器的端部的工作的算法。可以在[60]中找到有关这些卷积技术更普遍的细节。然而，我们的算法不会对常用分割双打块长度每隔一个块进行操作。我们的系统提供了一个特殊的块大小调理关于特定 PC 硬件性质，例如，高速缓存大小或特殊处理的结构，例如 SIMD（单指令多数据）。因此，最佳的卷积只添加第一块的时间延迟到该系统的延迟，所以它建议使用尽可能小的块长度。处理能力的量不是线性的总体滤波器长度和也由选择的开始块长度的限制。由于这个原因，测量完成，以确定不同的操作模式的处理器负载（见表格 1）。

表 1：低延迟卷积算法的 CPU 的负载



| Impulse response length | Number of sources | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 3 | 10 | 15 | 20 | 3 | 10 | 15 | 20 |
| | | (Latency 256 taps) | | | | (Latency 512 taps) | | |
| 0.5 s | 9% | 30% | 50% | 76% | 8% | 22% | 30% | 50% |
| 1.0 s | 14% | 40% | 66% | — | 11% | 33% | 53% | 80% |
| 2.0 s | 15% | 50% | 74% | — | 14% | 42% | 71% | — |
| 3.0 s | 18% | 62% | — | — | 16% | 53% | — | — |
| 5.0 s | 20% | 68% | — | — | 18% | 59% | — | — |
| 10.0 s | 24% | — | — | — | 20% | 68% | — | — |

## 5 系统集成

该 VirKopf 系统构成双耳合成和再现系统，视觉声学耦合，并且它被连接到 RAVEN 系统为室内声学模拟。整个系统的所有部件的布局在图 10 中示出这样它描述了分布式系统，用于在亚琛工业大学，在那里用户交互由六个摄像头跟踪的洞穴般的环境可听化。作为一种视觉 VR 机，双 Pentium 4 机 3 GHz 的 CPU 速度和 2 GB 的 RAM 使用（集群主机）。对于音频子系统 VR 的主机是双皓龙机 2 GHz 的 CPU 速度和 1 GB 的 RAM。房间声学仿真的速龙 2 GB 的 RAM 运行 3000+机器。这种硬件配置也用作测试系统的所有性能测量。作为音频硬件，一个 RME 　　落锤系统中使用，让声音输出与一个可扩展的缓冲区大小为 1.5 毫秒最小的延迟流。在我们的情况下，输出缓冲器大小被选择为 256 个抽头（5.8 毫秒）。所有 PC 之间的网络互联是一个标准的千兆以太网。

## 5.1 实时性要求

耦合实时系统的中央方面是延迟和用于通信的更新速率。为了获取所需的更新率的客观标准，它是强制性检查内洞穴般的环境中的典型行为，特别对于头部运动的类型和严重程度的位置或速度的变化。

在一般情况下，在 CAVE 状环境用户动作可以在三类进行分类。一类是通过以向

下和从一侧到另一移动并积累有关其结构特性信息识别由用户检查的固定物体的运动行为。第二类可以在运动中可以看出，当用户站在一个地方，并使用头部或身体旋转查看 CAVE 不同的显示面。可当用户在做两个，边走边在 CAVElike 环境东张西望地观察头部运动的第三类。主要，我们采用的典型应用程序可被归类为最后两个类别的情况下，虽然确切的用户运动轮廓可以单独不同。关于在虚拟环境中的典型的头部运动的理论和经验讨论仍在研究的受试者。

作为一个研究领域的研究中，我们记录了用户的头部运动跟踪数据，而在我们的虚拟环境互动。从这些数据，我们计算头部转动和平移的速度的大小，以确定用于室内声学模拟的要求。图 11（a）示出了平移速度的评估数据的直方图。从数据的偏差以下，平均平移速度是 15.4 厘米/秒，15.8 厘米/秒的标准偏差，并为 10.2 厘米/秒的数据值，比较图 11（c）所示。这表明，室内声学模拟的更新速度可能相当低平移的整体声音印象并不在附近发生大的变化（见[65]以获取更多信息）。举个例子，假设一个音乐厅的室内声场模拟，其中用于触发原始房间脉冲响应重新计算的门槛为 25 厘米（通常是排座位的距离的一半）。相对于用户的平移运动轮廓，重新计算必须大致完成每 750 毫秒捕捉运动的 70％左右。如果系统的目标是计算正确的图像源的运动的约 90％，这将有每 550 毫秒进行。的裸脉冲响应中包含的图像，它们的振幅和延迟的原始数据，而不是其在听者的坐标方向。慢慢更新的数据集表示，因此图像源的 roomrelated 云。转换成 3D 听者的坐标和卷积会快得多更新，当然，以允许直接的和光滑的响应速度。

CAVE 状环境允许用户直接在现场移动，例如，通过在显示表面和跟踪区域的边界的内侧行走。此外，间接导航使用户在风景移动几乎没有移动他的身体，但通过手传感器或操纵杆时指出隐喻。间接导航是强制性的，例如，建筑演练的虚拟场景通常比由洞穴般的设备本身所覆盖的空间大得多。对于间接导航最大速度具有为了避免在声学再现和感知假象或失真的限制。然而，间接运动期间，用户不倾向于移动其头部和整体感降低来评估模拟的正确性的能力。一旦用户停止，大约需要 750 毫秒如上所描述，计算右侧的过滤器为当前用户位置。我们提出，间接导航至 100 厘米/秒的最大速度的限制显示了良好的效果和用户接受的体验。

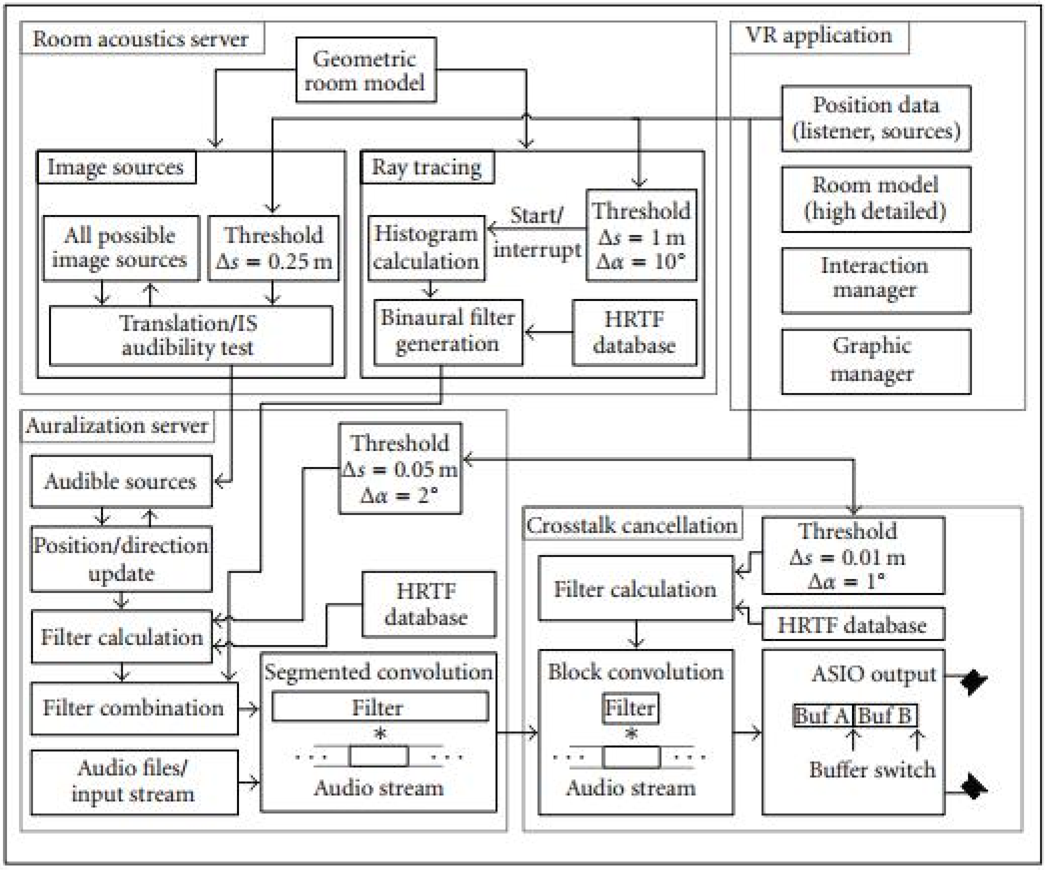
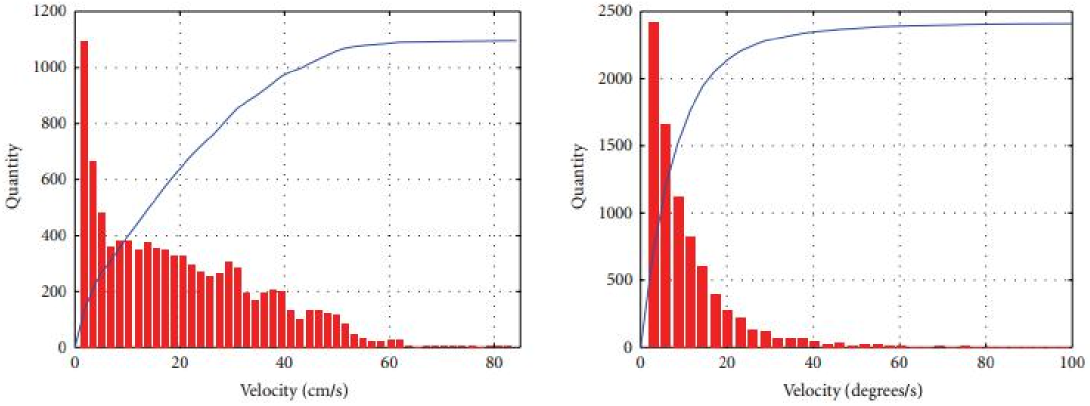


图 10：完全双耳可听化系统



(a)

(b)

| | $v_t$ (cm/s) | $v_r$ (deg/s) |
|---|---|---|
| $\bar{x}$ | 15.486 | 8.686 |
| $\sigma$ | 15.843 | 11.174 |
| $\tilde{x}$ | 10.236 | 5.239 |
| $x_{\max}$ | 84.271 | 141.458 |

(c)

图 11：平移（VT）的直方图，并在洞穴般的环境作用的用户的动作转动（VR）的速度。该蓝线表示测量结果的累计百分比。在（b）中，我们限定的上限为 100 度/秒为更好的可读性，（c）表示对测量的描述性统计

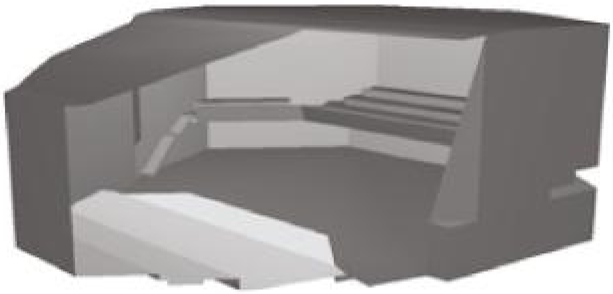


图 12：亚琛 Eurogress 国会会议中心音乐厅切多边形模型

除了平移行为，如图 11（b）示出了一个用户的头部运动的转动轮廓。峰角速度可高达每秒 140 度虽然这些是非常少见的。为旋转运动的平均值为 8.6 度/秒与 5.2 度/秒的 11.1 度/秒的标准偏差和一个数据值，比较图 11（c）所示。提供作为有关系统延迟研究的标准材料，例如，通过[66]或[61]，数据集显示比较的结果。

用户在声场头的取向是很关键的，因为反射具有用于在听者的坐 headrelated 脉冲响应来计算的。头旋转期间的 HRTF 的变化 ITD 可能导致两个滤波器一个显著相位失配。在交叉衰落从一个房间脉冲响应下一个，这些差异不应太大，因为这可能会导致可听梳状滤波器的效果。为了减少这些差别，过滤器改变每 1-2 度这里是必要的。为了精确的几乎所有可能的旋转速度，我们考虑对于每 10-20 毫秒为强制性重新计算的定时间隔。其结果是，块大小中配置音频处理硬件不应该大于 512 个样本，因为这在 44.1 kHz 的采样速率限制了可能的最小更新时间为 11.6 毫秒。

## 5.2 房间声学仿真性能

为了评估执行情况，并确定它的实时功能，一些实验测试系统上进行。对于一个现实的评估，亚琛 Eurogress 国会的音乐厅模型（体积约 1.5 万立方米）会议中心构建，这是在如图 12 所示。

在这方面的贡献呈现所有的结果都基于这个模型。

该模型是分别构成的 105 多边形和 74 的平面。尽管它保持相当简单，该模型 CON-tains 它们是声学的兴趣[67]房间的所有室内元件，例如，舞台，歪斜壁元件，并且栏杆。小的基元的细节被忽略并且由等效散射[68]表示。表面性质，即，吸收和散射系数通过标准化素材数据[69，70]中定义。

### 5.2.1 图片来源性能的方法

的计算时间的主要的声源，其各自的图像源的平移运动仅取决于图像源的数量。测量每 1000 大约 1 毫秒的图像源的平均计算时间。是需要的可听度测试的计算时间的主要部分。

为了给实现加速的一个更好的想法通过使用 BSP 树，蛮力 IS 可听性试验已经进行比较的目的得到落实。该算法测试每一个场景的十字路口上，而不是多边形的测试只有少数几个房间的一个 BSP 树结构的手段子分区。图 13 示出的测量计算时间指定 IS-可听

度测试的比较高达第二 IS 顺序的这两种方法。正如所料，穷举法的计算时间呈指数与指数越来越多的国际空间站由于搜索的复杂性下降至Ø上升，而基于 BSP 的方法只有一个相当线性增长计算时间的需求（日志 N）N 多的多边形。

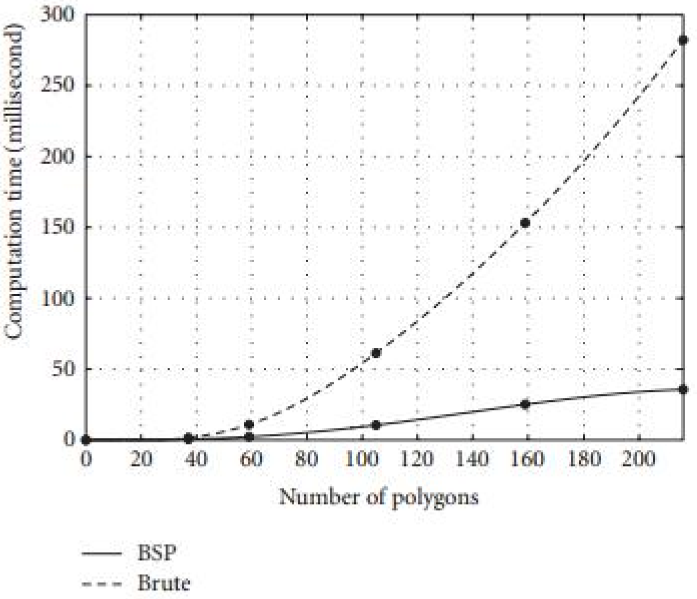
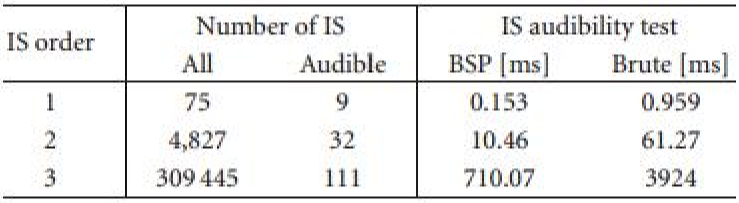


图 13：国际空间站可听度测试达到二阶国际空间站不同型号 Eurogress 国会这在他们详细程度不同，所需的运算时间比较（见[38]详情）。随着越来越多的多边形为模型的不同的详细程度的，ISS 的数目呈指数增长，这导致的计算时间为暴力破解方法呈指数增长。所述的计算时间需求 BSP-基于方法只生长线性由于搜索复杂的降幅高达 O（日志 N），N 多边形数量。

表 2：在 IS 可听试验的测定结果的比较



| IS order | Number of IS | | IS audibility test | |
|---|---|---|---|---|
| | All | Audible | BSP [ms] | Brute [ms] |
| 1 | 75 | 9 | 0.153 | 0.959 |
| 2 | 4,827 | 32 | 10.46 | 61.27 |
| 3 | 309 445 | 111 | 710.07 | 3924 |

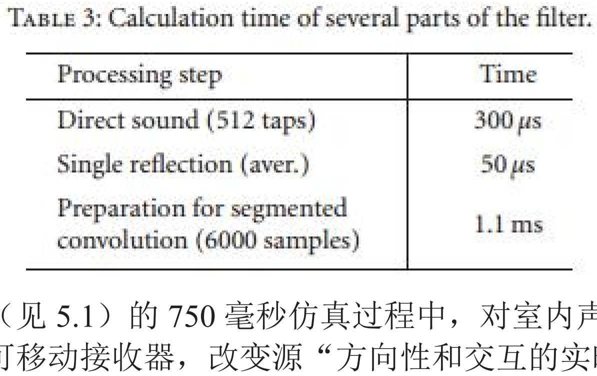


TABLE 3: Calculation time of several parts of the filter.

| Processing step | Time |
|---|---|
| Direct sound (512 taps) | 300 μs |
| Single reflection (aver.) | 50 μs |
| Preparation for segmented convolution (6000 samples) | 1.1 ms |

随着分配的时隙（见 5.1）的 750 毫秒仿真过程中，对室内声场模拟与所有的自由度，如可移动声源，可移动接收器，改变源“方向性和交互的实时能力达到约 320 000 的 IS 在运行时进行测试风景。应用这些约束的测量结果 IS 可听度试验（见表 2）使得与 Eurogress 模型实时能力的模拟到订购 3。

除了室内声学模拟的性能，滤波器的处理时间是非常重要的。在本节介绍的计算程序的所有时间测量我们的测试系统上执行。计算与 Eurogress 模型到第三顺序的图像源，

可听图象源可以在 6000 样本长度的对应于 136 毫秒的脉冲响应的第一部分找到。在这种情况下，一个源被放置在舞台上，和听众位于中间的空间。完整的滤波处理（不包括可听测试）在 6.95 毫秒完成。注意，滤波处理有不同的入口点。听者或源的旋转不会导致的可听源的重新计算，仅滤波器进行处理。

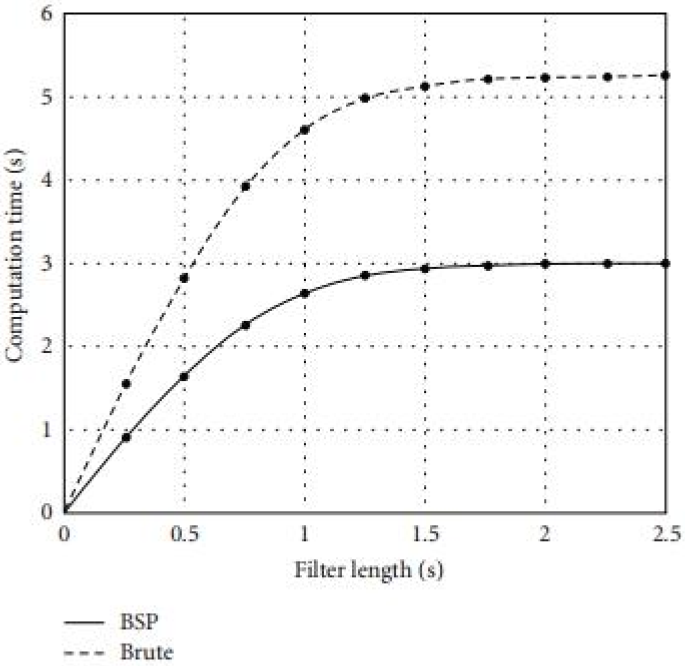


图 14：用于治疗与增加的长度的脉冲响应用 80000 射线为模拟的确定的所需的计算时间的比较

## 5.2.2 光线追踪性能

用于测量射线跟踪算法的性能，与 Eurogress 模型的所有材料通过单之一，以避免对结果不同的散射和吸收系数的影响取代。

正如上一节中，蛮力光线追踪算法已经实现，比较的结果与我们在我们的框架使用基于 BSP-方法。虽然蛮力方法有线性增长计算时间，也就是一个复杂度 O（N），N 多边形数量，基于 BSP-算法只对数随时间因搜索复杂性下降到 0 增长（日志 N）（见图 14，T <0.8 秒）。如果达到最小能量阈值一缕被终止。因此，这两种方法得到随着时间更快，因为越来越多的反射，分别是，越来越多射线“的能量和射线终止的损失。作为一个例子，该算法用于判定与 1 SECONE 的长度的脉冲响应的需要每 80 000 条（每频带 10 000 光线，前两个倍频程被跳过）约 2.6 秒的平均值。作为直线与所用的光线的数量的光线跟踪算法的处理时间增加，这些结果的比较是多余的。很明显，该算法能够应付实时要求，采用光线的小数字特别是当在第一次获得低分辨率直方图。如果侦听停留在一个地方的时间更长的时间，光线跟踪器可以更新与更多的光线直方图来获得更高的分辨率，并确定一个较长的脉冲响应。

## 5.3 网络

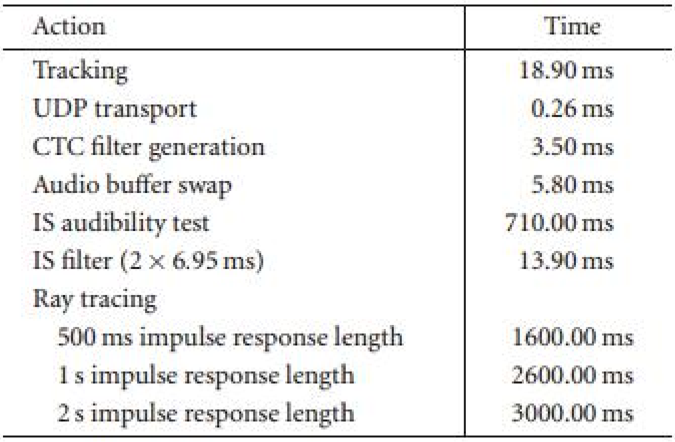
相对于定时，光学跟踪系统能够在 18.91 毫秒递送位置和用户的头部的方向和附加的交互设备向虚拟现实应用的空间的更新。该图是从所需的可视时间的总和的直接结果两个跟踪目标识别以及用于在网络链路将测量数据传输时间。对于必须有一个最小的延迟时间，并且不需要无线跟踪应用中，一个电磁跟踪系 统的使用可以减少延迟到包含约 5milliseconds。

然而，VirKopf 系统两种类型的更新消息的区分。一类低频状态的变化，如播放或停止特定声音指令交易。第二类型以高频率更新的声源和收听者的空间属性。对于第一类，可靠的传输协议用于（TCP），而后者则是在一个高频通过低开销可能不可靠的协议（UDP）的传输但为了获得网络的运输成本的估计，由 VirKopf 系统所产生的最大可能的 TCP 和 UDP 消息被从 VR 应用到 VirKopf 服务器多次发送，然后发送回。此往返传输时间取出并减半为单行程的测量。单行程的最坏情况时间被作为用于通过网络通信引入的整体成本的估计的基础。的平均时间为发送 TCP 命令为 0.15 毫秒±0.02 毫秒。在 TCP 通道的最坏情况下的传输时间接近 1.2 毫秒。UDP 通讯测定为 25 的声源 20 000 空间更新的表，从而为 0.26 毫秒±0.01 毫秒的表的发送时间。这似乎令人惊讶的是 UDP 通讯比 TCP 更贵，但这是从比较的空间更新（≈1 KB）的大数据包大小 TCP 命令尺寸（≈150 个字节）的结果。

## 5.4 整体表现

几个方面都被考虑到，得到完整的系统，多个子系统的性能，并行处理的组织中，网络传输的性能的概述，而且风景，即模拟房间（尺寸和几何的复杂性），源的速度，最后的使用者。更新房间声学模拟是该系统的最耗时的部分，并要求获得最佳感知性能的策略。图片来源和光线追踪在不同的 CPU 单独处理。光线跟踪处理的双耳滤波器将直接光线跟踪服务器上计算。该可听化服务器计算图像源过滤器并结合光线追踪过程中的所有过滤段。图 15 说明了图像源过滤的光线追踪和组合的一种可能的分割。应当提到的是，镜面部分的长度是房间依赖性。该光线跟踪中断点将基于所述监听器和源的移动速度进行调整。这意味着，该音频信号被滤波与室内脉冲响应的更新的第一部分，同时通过光线追踪已故部的生成仍在进行中。过滤器段进行更新将从完整的过滤器与 32 个样品≈0.72 毫秒的很短的斜坡被切断，并且新的段将被放置在具有相同的斜坡，以避免可听伪像。

表 4：几个子系统的性能测量结果的概述



| Action | Time |
|---|---|
| Tracking | 18.90 ms |
| UDP transport | 0.26 ms |
| CTC filter generation | 3.50 ms |
| Audio buffer swap | 5.80 ms |
| IS audibility test | 710.00 ms |
| IS filter (2 × 6.95 ms) | 13.90 ms |
| Ray tracing | |
| 500 ms impulse response length | 1600.00 ms |
| 1 s impulse response length | 2600.00 ms |
| 2 s impulse response length | 3000.00 ms |

由于所有这些因素的依赖性，更新时间不能被一般估计。因为这个原因，我们将给出一些详细的实施例相对于所述性能测量结果（见表 4 和 5）在几个部分制成的上方。应当注意到，所述图像源过滤器将在任何时间源或头部移动超过 2 厘米或接通多于 1 度，分别被更新。图像源过滤器会发出声响的来源目前的名单上计算（位置更新）。由此产生的过滤器只包含了一些错误的反思，这将可听试验之后被删除。因此，在脉冲响应的第一部分的镜面反射成为发声用正确的空间表示已经经过 35 毫秒（跟踪+ UDP 传输+ CTC 过滤一代一代过滤器+音频缓冲交换）。这也是一个收听者头部的旋转反应所需要的时间（见表 5）。

## 6 总结

在这种贡献，我们引入了一个相当复杂的系统模拟和室内声学可听化的实时性。该系统能够在任何类型的机箱的模拟房间声学声场没有任何扩散场条件的前提的。房间形状可以因此是极长的，扁平，耦合，或任何其它特殊性质的。表面特性，也可以自由地通过根据标准化材料数据使用波散射量选择。此外，该系统包括用于基于动态串音消除（虚拟耳机）单个用户的声场再现。该软件在标准 PC 硬件来实现的，需要没有特殊的处理器。性能（模拟处理时间，过滤器更新速率，跟踪器和声音硬件延迟）进行了评估，在的情况下充分考虑中等大小的音乐厅。

该系统的特定特征如下。

（一）它不是基于理想的扩散声场的任何假设，但两部分满房间声学模拟。镜面和脉冲响应的散射分量分开处理。除了能在低频小房间被处理任何类型的房间形状和体积。

（二）对于镜面反射和漫反射的量的决定仅仅是室温依赖性和纯粹基于物理声场方面。

（三）用户将只是涉及创建室 CAD 模型和吸收和散射的标准材料的数据。因此，可以使用的商业非实时仿真软件导入功能。该可听化是在执行的事实。

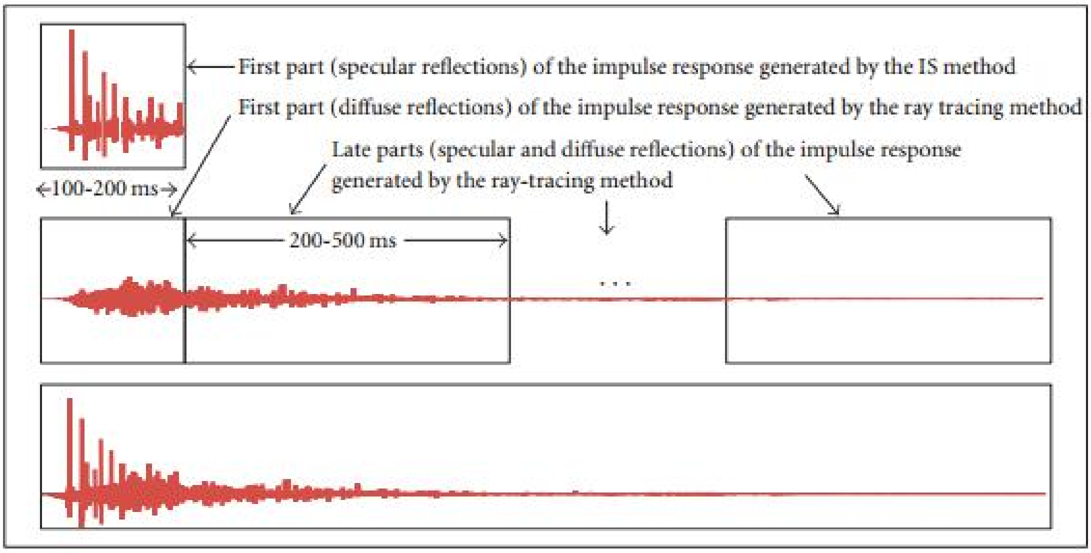
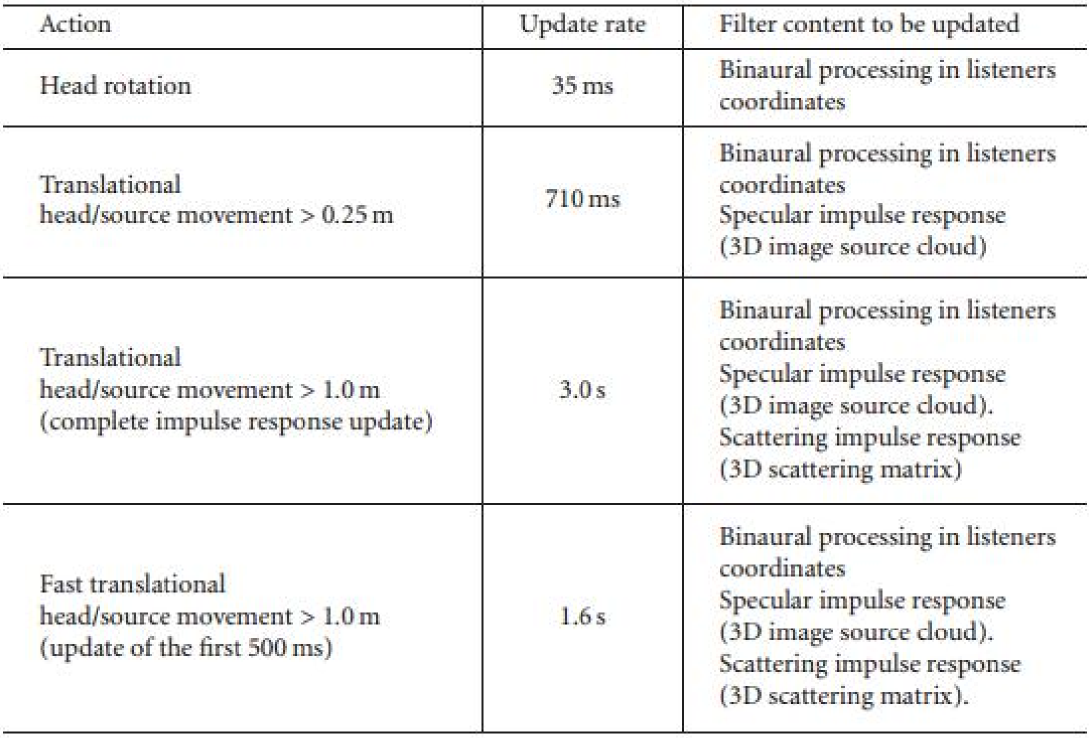

First part (specular reflections) of the impulse response generated by the IS method
First part (diffuse reflections) of the impulse response generated by the ray tracing method
Late parts (specular and diffuse reflections) of the impulse response generated by the ray-tracing method
←100-200 ms→
200-500 ms
...

图 15：过滤器（或过滤段），用于通过光线追踪和由图像源模型生成的脉冲响应的第一部分产生的一个耳朵的结合

表 5：更新间隔表 4 所示不同的模式以及基于所述测量头或源运动条件



| Action | Update rate | Filter content to be updated |
|---|---|---|
| Head rotation | 35 ms | Binaural processing in listeners coordinates |
| Translational<br>head/source movement > 0.25 m | 710 ms | Binaural processing in listeners coordinates<br>Specular impulse response (3D image source cloud) |
| Translational<br>head/source movement > 1.0 m<br>(complete impulse response update) | 3.0 s | Binaural processing in listeners coordinates<br>Specular impulse response (3D image source cloud).<br>Scattering impulse response (3D scattering matrix) |
| Fast translational<br>head/source movement > 1.0 m<br>(update of the first 500 ms) | 1.6 s | Binaural processing in listeners coordinates<br>Specular impulse response (3D image source cloud).<br>Scattering impulse response (3D scattering matrix). |

实时意味着用户不需要进行任何附加的任务。该系统将自动调整内在所有相关的运行参数，如分成镜面反射和分散零件和过滤器更新速率。

（四）该双耳脉冲响应的分量的处理被分离有关模拟本身，更新速率的可听化服务器，以及卷积处理。

（五）有关的更新速率和脉冲响应仿真的深度的决定是基于在 VR 系统的交互和用户的移动速度。

（六）中的脉冲响应，其延迟的正确性，并且其声音入射方向的正确性的信息的精度都只是根据在脉冲响应的相对到达时间。这与人的听觉系统关于本地化和回声延迟的能力的协议。是也应该在这里提到的是模拟深度和更新速率的系统参数不是由用户控制，但在系统中固有的处理。这种处理方式将在直达声和反射第一的最早期创建完整的复杂性和可听化精确。渐渐地，声能会被转移到脉冲的散射分　　量。精度和更新速率降低，由限制动机由于掩蔽效应音质。该系统是开放的进一步延伸相对于声音的衍射和隔音效果。

房间声学仿真软件的实时性能是通过引入虚拟环境的交互可听化的灵活框架的实现。场景的概念的图表为通过自主地操作子场景的有效和灵活的联动

所谓门户装置已被并入现有的框架，并与下面的 BSP 树结构用于处理几何问题很快相结合。使用该框架的提供了用于既施加算法显著减少的计算时间（确定性图像源和一个随机射线追踪）的可能性。尤其，图象源方法通过引入空间数据结构的改善作为门户状态可以被利用，使得能够显着减少将要处理的图像的来源的数量。

快速低延迟引擎确保了无论其全长的脉冲响应将通过单声道音频材料的过滤后 5.8 毫秒（块长度 256 个样本）进行审议。关于现代处理器扩展的优化使的呈现，例如，10 个信息源与 3 秒（132 000 个抽头）的长度或 15 的来源与 2-第二长度的过滤器。

双耳音频信号的再现是由一个动态串扰消除系统，而不会限制用户的运动提供。本系统作为一个虚拟耳机无需穿物理耳机提供信道分离。

千兆以太网用于连接视觉呈现系统和音频系统。视觉的 VR 系统发送控制命令以及所述头部的空间的更新和来源。该控制命令（例如启动/停止），这样的变化与下一个声音输出块担任了紧张的音频视频同步将在音频服务器后，0.15 毫秒考虑。

## 7 展望

尽管整个系统的性能良好，存在具有待研究的许多方面。以进一步提高室内声学模拟的质量，如隔音和衍射物理效应将被并入现有的算法。此外，施罗德频率以下的频率的模拟可以通过快速和动态的有限元法（FEM）-solver 的手段来完成。现有框架已经打开采取这些现象考虑，相应的算法已时才被执行。目前，模拟软件在第一版本作为自包含的稳定的基础来实现。因此，优化算法是必要的，以进一步提高其性能，特别是注重在并行处理的计算。位置预测可能是减少的位置的偏差的可能性，在过滤器进行计算，与实际的听者的位置。

初步听力测试表明，所产生的虚拟源可以在低的错误率进行本地化。房间声学仿真被认为是合理的，并匹配所产生的视觉形象。在未来，更测试将完成评估的限制

更新速率和源的数量。基于感知减少如陈述在例如，[71，72]是也降低了处理成本的一个有趣的方法，并且将在未来被考虑。

## 致谢

笔者想在编程工作的一部分，也是托斯滕 Kuhlen，安德烈亚斯弗兰克和马克 - 奥利弗 GULD 的支持，并讨论感谢弗兰克 Wefers，希尔马德穆思，和菲利普渣为他们的承诺。此外，由于 DFG 资助项目（DFG 项目“虚拟耳机，”2004- 2006 年）的一部分。最后，作者要感谢匿名审稿人对于扩展工作，这帮助了很多需要提高这方面的贡献。

## 参考文献：

[1] D. R. Begault，“挑战成功实施的 3-D 声”杂志的音频工程协会，第一卷，第 864-870，1991。

[2] MNAEF，OStaadt 和格罗斯在 ACM 研讨会论文集虚拟现实的软件和技术“为沉浸式虚拟环境，空间化音频呈现”（VRST '02），第 65-72，香港，2002 年 11 月。

[3] C.克鲁兹内拉，D J.桑丁，T. A. DeFanti，R.五肯扬和 J. C.哈特，“洞穴：视听体验自动虚拟环境” ，美国计算机协会通讯，第一卷 35，6，第 65-72，1992。

[4] D. A. Burgess 和 J. C. Verlinden，“为空间音频服务器的体系结构”，在虚拟现实系统大会（秋季'93），纽约，NY，USA，1993 年 11 月提起诉讼。

[5] JD Mulder 和 EH Dooijes，“在图形应用空间音频”，在科学计算可视化，M. GOBEL，H.穆勒和 B.城市编，第 215-229，施普林格，维也纳，奥地利，1994。

[6]休伦湖，2005 年，http://www.lake.com.au/。

[7] L. Savioja，虚拟声学建模技术，博士论文，科技，芬兰赫尔辛基，1999 年 12 月的赫尔辛基大学。

[8] L. Savioja，J. Huopaniemi，T Lokki 和 R.弗吉尼亚州的 ANEN，“创建交互式虚拟的声学环境”杂志音频工程协会，第一卷，第 675-705，1999。

[9] T.芬克豪泽，P.敏，和我 Carlbom，“实时声学建模分布式虚拟环境中，”诉讼中的计算机图形和交互技术（SIGGRAPH '99），第 26 届年度会议。365 -374，洛杉矶，加利福尼亚州，美国，1999 年 8 月。

[10] R. L.风暴“Npsnet-3D 音效服务器：听觉通道的有效利用”，1995 年。

[11] H. Kuttruff，室内声学，爱思唯尔科学出版社，纽约，NY，USA，第 4 版，2000。

[12] J. B. Allen 和 D. A.伯克利，该杂志美国卷声学学会“为有效地模拟小房间声学，图像法”。 65，没有。 4，第 943-950，1979。

[13] J. Borish“的形象模型扩展到任意多面体”的美国杂志，第一卷声学学会的。75，没有。 6，第 1827 至 1836 年，1984 年。

[14] B.-I. L. Dalenback，“根据漫反射和镜面反射的统一处理室的声学预测”的美国杂志，第一卷声学学会的。 100，没有。 2，第 899909，1996 年。

[15] P.-A.福斯贝里，“全离散光线追踪，”应用声学，第一卷。 18，没有。 6，第 393-397，1985。

[16] T.芬克豪泽，N. Tsingos，Carlbom，等人，“交互式建筑声学光束跟踪方法”的美国杂志，第一卷声学学会的，第 739-756,2004。

[17] G. M.奈勒“ODEON 另一混合室的声学模型，”应用声学，第一卷，第 131-143，1993。

[18] U. M.斯蒂芬森，“量化锥体束跟踪室内声学和噪音引入预后，一种新的算法”声学学报美国与声学，第一卷，第 517-525，1996。

[19] D.面包车 Maercke，在声学上的第 12 届国际大会论文集（ICA 1986 年），第二卷“采用几何模型，在时间和频域声场模拟”，多伦多，加拿大安大略省，1986 年 7 月，造纸 E11-7。

[20] M. Vorlander，“采用了全新的组合声音粒子图像源算法客房瞬态和稳态声传播的模拟”的美国杂志，第一卷声学学会的，第 172-178，1989。

[21]一，博克，声学学报美国与声学，第一卷“房仿真软件，第二轮罗宾对室内声场计算机模拟，比较”，第 943-956，2000。

[22] M. Vorlander，在声学上的第 15 届国际大会论文集（ICA '95）“关于室内声学计算机模拟，国际循环赛”，页 689-692，挪威特隆赫姆，1995 年 6 月。

[23] H. Kuttruff“，为衰减常数的计算与漫反射边界上进行简单迭代计划在外壳”的美国杂志，第一卷声学学会的。 98，没有。 1，第 288-293，1995。

[24] C. L. Christensen 和 J. H.林德尔在论坛 Acousticum，匈牙利布达佩斯，2005 年的“结合粗糙度和衍射效应，一种新的散射法”。

[25] R。亨氏，“双耳房间基于与另外的统计方法的图像源模型模拟，以包括壁的漫射声散射和预测混响尾巴”应用声学，第一卷，第 145-159，1993。

[26] Y. W.林“，在室内使用的 3 个反射建模方法比较声学计算机模型”的美国杂志，第一卷声学学会的，第 2181 至 2192 年，1996 年。