文章编号：2096-1472(2019)-08-14-03
DOI:10.19644/j.cnki.issn2096-1472.2019.08.005

一种基于WebVR实现3D音效可视化的方法

金 益，张 量
(苏州市职业大学计算机工程学院，江苏 苏州 215104)

摘 要：由于VR场景中用户可以自主选择观看的方向和角度，为了获得最佳的体验效果，VR音频技术要求通过声音辨别方向、距离等反馈更多的空间环境状态。本文主要论述了一种基于WebVR实现音频空间可视化的方法，核心是运用了Web Audio API的PannerNode和AudioListener两个对象处理音频源，其中Three.js提供了完善的音频空间化支持。最后通过VR Audio的案例进行测试，实验表明根据用户的位置和旋转矩阵，可更新音频空间信息。

关键词：WebVR；音频空间化；Web Audio API；音频源
中图分类号：TP391 **文献标识码**：A

A Method of Visualizing 3D Sound Effect Based on WebVR

JIN Yi,ZHANG Liang
(*Dpt. of Computer Engineering,Suzhou Vocational University,Suzhou* 215104,*China*)

Abstract: As users can choose the direction and angle of view independently in VR scene,VR audio technology requires more spatial environment status through sound discrimination direction,distance,etc.in order to gain optimum experience.This paper mainly discusses a method of audio spatial visualization based on WebVR.The core of this method is to use Web Audio API's PanerNode and AudioListener to process audio source,in which Three.js provides refined support to audio spatialization.Finally,a VR audio case is tested.The experiment shows that the audio spatial information can be updated according to the user's position and rotation matrix.

Keywords: WebVR;audio spatialization;Web Audio API;audio source

1 引言(Introduction)

在虚拟现实应用的开发中，除了图形视觉渲染之外，音频处理也是重要的一个环节，最佳的3D音效能使用户达到身临其境的效果[1,3]。VR音频的输出硬件主要是耳机，根据音频源与场景之间的关系，可将VR音频分为两类：静态音频和空间化音频[1]。

静态音频作用于整个VR场景，可简单地理解为背景音乐，音频输出是静态的，比如雨滴声、闹市声等充斥整个场景的背景音效。对于此类音效的开发，我们可以简单地使用HTML5中的<audio>标签进行循环播放。空间化音频作用在空间的实体上，具有发声体和听者的位置关系，音频输出会根据发声体与用户的距离、方向动态变化，它模拟了现实中声音的传播方式，在虚拟场景中，通过调节音频的振幅来描述发声体与听者之间的距离，再通过调节左右通道之间的差异，控制左右耳机喇叭输出，来描述发声体相对听者的方位，具有空间感[2]。从发声体与用户两点间的距离来看，如距离越远，音频音量(振幅)应越小；从发声体与用户的方向来看，如发声体位于听者左侧，则音频输出的左声道应比右声道音量大。VR的数字内容缺少了空间音频，甚至音画不同步，那用户的临场感将大打折扣。为了实现以上复杂的空间化音频，我们提出一种基于WebVR的方法。WebVR是指使用浏览器体验VR的方式，如今已经成为了一种开放标准，它提供了Web Audio API接口，一个功能强大的音频处理系统，允许我们在浏览器中通过js来实时控制处理音频，比如音频可视化、音频混合等，让原本较为复杂的音频空间化处理变得简单。

2 音频开发流程(Process of audio development)

我们首先用WebVR提供的Web Audio API处理一个简

基金项目：本文系2019年江苏省大学生创新创业训练计划项目资助（项目编号201911054011Y）；2018年苏州市职业大学创新项目资助（SVU2018CX12）；江苏省高校哲学社会科学重点建设基地吴文化传承与创新研究中心（2018ZDJD-B018）项目成果；苏州市职业大学“青蓝工程”青年骨干教师资助；江苏省“青蓝工程”优秀教学团队资助；江苏开放大学办学体系“十三五”2019年度科研规划课题（资助编号19TXYB-05）.

www.rjgczz.com

单的音频。Web Audio API提供了非常丰富的接口让开发者在web平台上实现对音频进行处理，底层由c++引擎提供支持与优化[3]。利用Web Audio API，开发者能够对音频文件拥有更高的处理权限以及内置相关的音频专业效果处理。Web Audio API使用户可以在AudioContext中进行音频操作，具有模块化路由的特点。一个简单的音频处理流程需要分为五步，如图1所示。

（1）创建音频上下文。

（2）在音频上下文里创建源。例如 <audio>、振荡器、流等。

（3）创建效果节点，例如混响、双二阶滤波器、平移、压缩。

（4）为音频选择一个目的地，例如系统扬声器。

（5）连接源到效果器，对目的地进行效果输出。

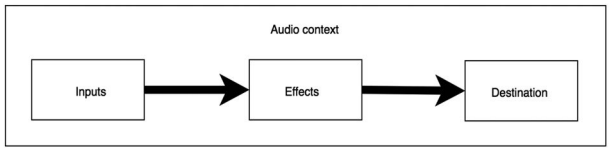

图1 音频处理流程

Fig.1 Flow chart for audio processing

Web Audio API处理流程可以比喻成一个加工厂对声源的加工，这个加工厂由多个加工模块连接而成，音频源经过一系列的处理加工后，被输送至扬声器。其中Audio Context代表了一个音频加工厂控制中心，是处理音频的核心对象[1]，所有处理连接过程都由audio context管理，负责各个音频结点的创建和组合，通过new AudioContext()的方式创建，如图所示，Audio Context控制Input节点的生成和Destination节点的生成，同时控制着Input节点与Destination节点之间的连接。AudioNode是音频结点，则是加工厂的加工模块，按照功能可分为三类：输入结点、处理结点、输出结点。每个结点都拥有connect()方法连接下一个结点，将音频输出到下一个模块。输入结点主要负责加载解码音频源，比如获取二进制音频源的BufferSourceNode、获取<audio>音频源的MediaElementSourceNode等[4]；处理结点主要对音频数据进行计算处理，比如处理音频振幅的GainNode等；输出结点则将音频输出至扬声器或耳机，AudioContext.destination便是默认的输出节点。在Web Audio API处理音频流程的基础上，我们在WebVR中实现音频的空间化处理。

3 空间化音频的实现(Implementation of spatialized audio)

3.1 设置AudioListener对象

空间化音频的实现主要通过AudioListener和PannerNode结点配合，这两个对象可以根据空间方位信息动态处理音频源，并输出左右声道。其中AudioListener对象代表三维空间中的听者(用户)，通过AudioContext.listener属性获取；PannerNode对象指的是三维空间中的发声体，PannerNode可以设置音频源的方位(上下、左右、远近)从而在听觉上产生空间的感觉，通过AudioContext.createPanner( )创建。我们需要初始化这两个对象，并将空间方位信息作为输入参数动态传给它们。设置PannerNode对象的代码如下所示[4](使用three.js进行开发)：

```
const myAudio=document.querySelector('audio');
const audioCtx=new AudioContext(); //创建音频上下文
const source=audioCtx.createMediaElementSource(myAudio);
const panner=audioCtx.createPannerNode();
panner.setPosition(speaker.position.x, speaker.position.y,speaker.position.z); //将发声体坐标传给PannerNode
source.connect(panner); //将输入结点连接到PannerNode处理结点
panner.connect(audioCtx.destination);
source.start(0); //播放音频
```

3.2 设置PannerNode对象

在实际的VR应用中，带着VR头显的用户最多有6Dof，即为用户的头部和身体均做上下、左右、前后回转动作。我们需要将这6Dof的信息传入AudioListener，由它为我们处理音频数据。对于用户位置数据，AudioListener提供了三个位置属性：positionX、positionY、positionZ，它分别代表听者当前位置的xyz坐标，我们可将用户在场景中的位置(一般用camera的position)赋值给这三个属性。

除了传入用户的位置，我们还需要将用户的视角方向信息传给AudioListener，具体是给AudioListener的Forward向量三个分量forwardX、forwardY、forwardZ和Up向量三个分量upX、upY、upZ赋值。Forward向量沿着鼻子方向指向前，默认是(0,0,−1)；Up向量沿着头顶方向指向上，默认是(0,1,0)。

在VR场景中，当用户转动头部改变视角时，up向量或forward向量会随之改变，但两者始终垂直。于是我们得到两个公式：

Up向量=Camera.旋转矩阵×[0,1,0]

Forward向量=Camera.旋转矩阵×[0,0,−1]

参照上方公式，这里的camera指代用户的头部，通过camera.quaternion获取相机的旋转(四元数)矩阵，与初始向量相乘，得到当前Up向量和Forward向量。

4 WebVR实现音频角色(Implementation of audio roles via WebVR)

在VR场景中，根据音频的发起方和接收方可以分为两个角色：Speaker发声体与Listener听者，即用户。一个VR场景音频角色由一个Listener和多个Speaker组成，于是笔者将前述的PannerNode对象和AudioListener对象进行独立封装，整合为Speaker类和Listener对象。

4.1 Speaker类实现

Speaker类代表发声体，主要完成以下功能：

(1)始化阶段加载解析音频源，创建并连接输入结点、处

www.rigcz.com

理结点、输出结点

(2)提供update公用方法，在每一帧中更新PannerNode位置。

代码实现过程如下：

```
update(position) {
const {panner}=this;
panner.setPosition(position.x,position.y,position.z);}   //将发声体坐标传给PannerNode_
loadAudio(path) {
//使用fetch请求音频文件
return fetch(path).then(res=>res.arrayBuffer());}
async _processAudio() {
const {path,ctx,source}=this;
try {
const data=await this._loadAudio(path);
//异步请求音频
const buffer=await ctx.decodeAudioData(data);//解码音频数据
source.buffer=buffer;//将解码数据赋值给BufferSourceNode输入结点
source.start(0);} //播放音频
catch(err) {
console.err(err);}
```

以上语句代表初始化的流程略有不同，这里使用的是fetch请求音频文件，通过BufferSourceNode输入结点解析音频数据。update方法传入发声体position，设置PannerNode位置。

4.2 Listener对象实现

Listener对象代表听者，提供update公用方法，在每帧中传入AudioListener的位置和方向。我们只是简单地将AudioListener作一层封装，update方法传入camera的position和四元数矩阵，设置AudioListener位置、方向。具体实操过程中，我们用两条关键语句let forward=new THREE.Vector3(0,0,-1)和let up=new THREE.Vector3(0,1,0)计算当前listener的Forward向量。

5 WebVR场景应用(WebVR scene application)

我们用一个简单的音频可视化案例进行测试，把Listener对象和Speaker类引入到WebVR场景应用中。VR场景我们依然用Three.js完成。首先构造函数先初始化场景、相机和渲染器；其次在渲染之前调用start方法，在start方法里我们为场景创建3d物体；最后，调用renderer.animate(this.update)开启动画渲染，update方法里可动态操作物体属性[5]。具体我们通过构建跃动的柱状条辅以动画，为了便于观察，我们将柱状条的正面用绿色贴图，反面用棕红色贴图，动画过程中进行了一些角度和镜头伸缩方面的设置，并调用Speaker和Listener的update方法，我们截取了音频空间柱形图的前视图、顶视图、旋转45度，翻转180度时的图形效果，表明当传入用户要求的位置和旋转矩阵，以此可更新音频空间信息。测试结果如图2所示。

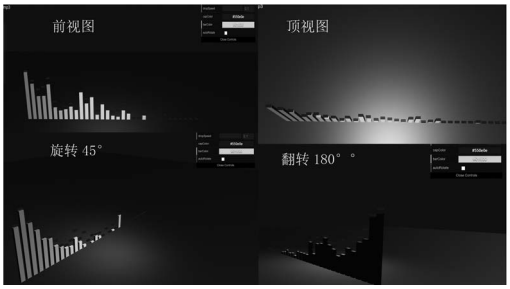

图2 WebVR应用结果

Fig.2 Results of WebVR application

6 结论(Conclusion)

随着硬件和软件技术的发展，产业界对VR用户体验产生了重大期望。目前，国外的Facebook、谷歌、火狐浏览器和国内百度已推出支持WebVR浏览器的版本[6]，微软也宣布将推出自己的VR浏览器，随着后期5G网络极速时代的到来以及HMD头显的价格和平台的成熟，WebVR的体验方式将是革命性的[7]。我们可通过现代浏览器进一步降低VR内容的技术创作成本及门槛.3D音效作为VR体验不可或缺的一部分亦是如此。通过WebVR API接口让原本较为复杂的音频空间化处理变得更为简单，3D音效将为VR场景的体验带来更为身临其境的效果。尽管WebVR仍然不太稳定，比如某些情况下会有头显设备跟踪丢失的情况，我们会在后期的实验中不断改进和优化。

参考文献(References)

[1] Clayton Elieisar(克莱顿).VR空间里的交互式3D交响乐设计及其传播效果研究[D].中国科学技术大学,2017.

[2] 刘金广,孔银昌.虚拟场景中的3D音频的分析与实现[J].电子与电脑,2010(01):94-97.

[3] Quan Ye.Implementation of WebVR-based Laboratory for Control Engineering Education based on NCSLab Framework[A].中国自动化学会控制理论专业委员会.第36届中国控制会议论文集(E)[C].中国自动化学会控制理论专业委员会:中国自动化学会控制理论专业委员会,2017:6.

[4] 刘展海,何伟,刘展航,等.基于Three.js下的WebVR开发[J].科学技术创新,2018(09):82-83.

[5] 许昊天.WebVR大规模三维虚拟场景轻量对等传输关键技术[J].电脑知识与技术,2019,15(03):215-217.

[6] 林定,黄国新,徐颖.一种基于WebVR的网络数据三维树形可视化[J].系统仿真学报,2018,30(07):2736-2743;2752.

[7] 杨晨彬,高瞻.基于WebVR的医学体数据可视化[J].计算机应用与软件,2019,36(05):101-107.

作者简介：

金 益(1980-)，女，硕士，副教授.研究领域：虚拟现实VR应用.

张 量(1982-)，男，博士，讲师.研究领域：计算机应用技术.

www.rjgzz.com