

基于实时引擎的全景视频播放系统开发

北京电影学院声音学院 刘辰

【摘要】随着虚拟现实（VR）技术的成熟与普及，全景视频、VR游戏等概念与相关产品得到了长足发展。尽管全景视频拍摄和制作在工艺上逐渐成熟，但是全景视频在视频和音频的还放环节仍然存在缺少规范和灵活性的问题。本文通过开发一种基于实时引擎Unity的全景媒体播放系统，能够实现在还放视频的同时解决声音还放遇到的一些问题，为声音设计师提供更多实现手段和思路。

【关键词】全景视频 VR Unity Ambisonics

【中图分类号】J966

随着电影技术不断的成熟，人们也在不断探索影像表现的新形式。近年来VR（虚拟现实）技术的发展，使全景视频作为新的电影表现形式的可能性受到大家的关注。智能手机、Gopro相机等数码产品的出现更是降低了影像的拍摄和制作成本，使得小团队或者个人也能够便捷地制作全景视频内容。探索一种简易、灵活的全景媒体播放系统，解决全景视频和音频的不同播放需求，即是本文的主要研究目标。

一、全景媒体的发展

全景视频的出现最早可以追溯到上世纪四十年代末的“西尼拉玛电影”，属于宽银幕电影的一种。1952年在美国百老汇剧院上映的《这是西尼拉玛》可以视为早期的全景拍摄尝试。尽管水平视角只有146°，且因为成本过高没有得到大规模的普及，人们对于全景视频的探索却没有因此停下步伐。

到了70年代出现的“球幕电影”（穹幕电影），通过对鱼眼镜头的使用，半球形的银幕使观众在观看电影时，整个画面能够布满球体。1994年北京科技馆引进了我国第一座“球幕电影院”，并首映了美国影片《大峡谷》。“全景视频”正式走进了大众的

生活中。时至今日，我们也可以在一些大型商圈、娱乐设施、文化科技馆中享受类似的全景内容体验。

表1 国外主流虚拟现实硬件概览表

	Oculus Rift	HTC Vive	Google Cardboard	PS VR
开发时间	2013	2015	2014	2016
开发相关公司	Facebook Oculus	HTC Valve	Google	Sony
支持内容	VR游戏、全景视频、其他相关应用			
相关描述	领先业界的虚拟现实显示器，支持多方 开发平台	头部显示 器+操作手 柄，支持 Steam 平台 的 VR 游戏	盒状、可折叠、 组装简易，为有 3D 显示的智能 手机搭配提供 虚拟现实体验	主要使用 方式为体 验其主机 平台 PS4 上 的 VR 游戏

随着上世纪末互联网时代的到来，国内外如YouTube、优酷等视频内容播放平台纷纷崛起，丰富了人们的娱乐生活。人人都可以制作独特的视频内容并上传到网络平台进行分享。前两年以YouTube平台首度支持360°视频的上传与播放为起点，全景视频的内容产量也随之大幅增加。同时随着主流VR硬件设备的普及（表1），人们可以更加方便地获取VR游戏、全景视频或其他相关的内容。用户只要带上耳机或VR设备，拖动鼠标或是转动身

体方位，即可实现实时的全景试听体验。

随着数码设备的发展，全景视频的应用也逐渐广泛，包括但不限于导航软件的实景预览，家居、景观等行业的模拟观看，甚至在教学演示中都能看到它的身影。电影从业者对它的关注点则放在了全景视频给影视业界带来的冲击与迭代变化上。在这个过程中也诞生了许多作品与尝试性的内容：例如2014年，曾拍摄《速度与激情》的华裔导演林诣彬拍摄了实验性的全景视频短片《HELP》；2016年，导演罗布麦克莱伦（Rob McLellan）推出了全球首部限制级“VR”电影《ABE VR》；北京威锐影业同年发布了首次采用全景拍摄制作的实验电影短片《全侦探》等。VR立体影像在距离空间上的“补全”为影像的沉浸感、表现力增添了新的可能，更在影像的各个层面提出了诸多新问题。如在紧贴观看者的近距离影像空间中，观看者的肢体无法如同在传统影像媒介中一般作为假定性存在而忽略。尤其是VR游戏中，观看者的肢体等同于游戏中的角色，观看者的手脚甚至需要与影像中的内容互动。而视听语言层面的根本性问题，则需要更深入、理性的分析。

二、实时引擎

实时引擎，是在基于电脑实时运算处理信息的基础上，通过一系列已编写好的系统程序实现对影像、图片、音频等数据进行实时处理的综合平台。常见于影视、游戏开发、交互应用等领域，并凭借在代码上的开源性和所集成内容的综合性受到开发者和设计师们的欢迎。也就是说，设计师不必从零开始写代码，而是利用已有的工具快速进行创作，方便实现各种功能。以Unity为例，它是由Unity Technologies开发的一款开源的综合游戏开发工具，它支持C#等主流编程语言，体量较小并支持多平台发布，十分适用于中小型游戏开发或个人进行交互设计。本文中涉及的系统开发与测试均通过Unity完成。

三、全景媒体的声音还放与存在的问题

全景视频中，除了画面外，音频的制作与还放也作为全景媒体的一部分影响用户的体验。现在的

绝大多数情况下的全景视频体验通常使用耳机进行还放，因此本文将列举几种主要的全景视频声音还放方式。

传统的立体声显然是最常见的，是被几乎所有软硬件、视频播放器支持的还放方式，也是VR设备普及之前球幕影院等大型设施通常使用的还音方式。立体声虽可以表现声音的运动感，但缺点是声像无法跟着头部实时转动，定位感也不够清晰。在视频是可旋转的全景内容时，离VR概念中的“沉浸感”便相去甚远。

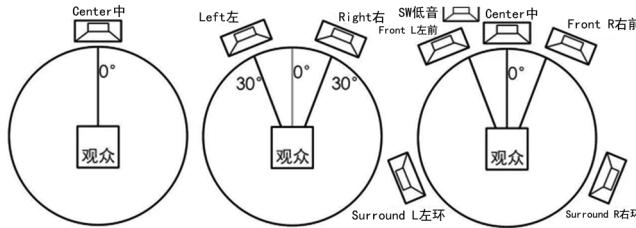


图1 影院单声道、立体声、5.1环绕声简略声场示意图

人头录音也被称为双耳录音（Binaural Recording），是一种基于HRTF（Head Related Transfer Functions头相关传递函数）的，使用模拟人工头和仿真外耳道模型来获取模拟人耳听觉声音效果的录音方式。区别于普通立体声的是，使用人



头录音的优点是会在水平方向获取较为清晰的声音定位，但是对声音高度信息的定位仍不够明确。另外，单纯的人头录音也无法满足头部转动产生实时变化的需求。

Ambisonic，又被译作“声场环绕声”，是1970年代在英国国家开发公司（BNRDC）的赞助下开发的一种球面环绕声技术。这项在当时具有前瞻性，却意外地在商业化过程中失败的技术，就是在这样的背景下重新回到大众的视野并受到关注和推广。一般地，Ambisonic被理解为MS制式立体声的三

维扩展,因为除了水平面之外,它还覆盖了听者上下方的声源。它使用一定方式组合的多个传声器阵列来记录整个全方位的球形声场信息。

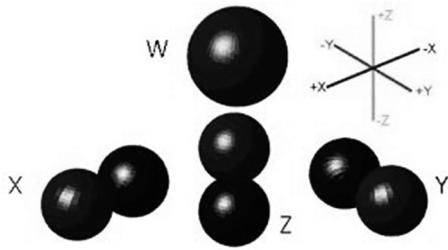


图3 一阶 Ambisonic 声场分解图

Ambisonics 技术能够让音频声像跟随头部旋转,并且凭借独特的编解码可以被作为一种音频文件格式来储存和播放,与人头录音相比,它在声音定位、编解码方面更加灵活清晰。数字视频文件播放的音轨通常是经过混底输出的单声道、立体声、5.1声道甚至是 ambisonics 的音频文件,交由媒体播放器、影院音响等进行还放。

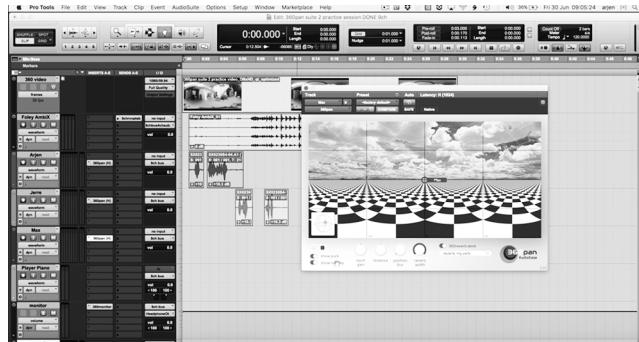


图4 audioease 360pan suite 插件允许声音设计师在工作站中对全景视频的声音进行声像制作并实时监听,支持输出 ambisonics 格式

对主要依靠头戴显示器和耳机观看全景视频的用户来说,立体声、环绕声制式的声音是按照特定的还放通道设计,声像无法随转动头部变化。虽然可能也适用于一些特定的内容,比如景观类的全景内容可能单纯地用立体声音或者音效铺底就足够。但是对全景的动画、电影这些有情节的、音响表达比较复杂的内容来说就稍有灵活性上的不足。使用 Ambisonics 固然可以实现声场的沉浸感和声像实时“转”的效果,但现有播放器却不允许它像环绕声那

样,在制作端对播放端进行配置。另外,一些用到实时技术的声音设计需求也无法在一条固定音轨上体现。这就需要开发一种系统来尽可能满足全景视频不同需求的声音设计与还放。

四、基于实时引擎的播放系统开发

本文主要通过游戏引擎 Unity 来进行开发和功能实现。首先,Unity 作为一款成熟的游戏引擎,体量适中,也常用于综合媒体等领域,它可以快速进行功能上的实现和调试,帮助设计师省去大量代码层面的操作,并且输出程序支持多平台,界面友好;其次,Unity 支持直接播放 H.264、PCM 等主流编码格式的视频音频文件,包括.mp4、.mov 等封装格式的视频和.mp3、.wav 的音频,同时 Unity 也支持对音频进行 vorbis、ADPCM 等编码的压缩,适合处理主流格式的全景视频;最后,Unity 能对包括 Ambisonics 在内的不同格式和通道的音频进行灵活的配置,通过调用脚本,或者集成第三方音频中间件的方式,来实现符合声音设计需求的音频播放规则。

本文使用北京电影学院动画学院学生的 VR 动画短片作品《离你一个头盔的距离》作为实验素材,视频是 3840×1920 的 mov 格式,音频为 48kHz、24bit 的四声道 B-format 文件。

为保证还放的流畅性,本次的视频素材尽可能使用了体量较小的格式和压缩编码。通过调用“镜头移动”和“材质透视”这两个 Unity 内的常用脚本来实现全景视频的播放:

(1) 建立一个新的工程后,新建一个 script (脚本),输入“镜头移动”的相关代码后,控制 Unity 内的“摄像机”作为第一人称的玩家视角能够随鼠标或 VR 设备的体感控制随意转动(代码可在任一开源网站上搜索获得);

(2) 将“材质透视”代码设置为 Unity 内部物体材质的 shader (阴影) 并把它和全景视频素材同时挂载到一个球形对象中,同时把主视角摄像机的坐标设置在球形对象的中间。这个过程模拟了人在观看一个 360°无死角的全景视频时与画面的位置关系。如此设置完毕后,在球面上挂载一个带有该视

频的 videoplayer 组件，点击 unity 界面上方的“播放”按钮，即可实现全景视频的基础播放和调试功能。

```
// Upgrade NOTE: replaced
'UNITY_MATRIX_MVP,*' with
'UnityObjectToClipPos(*)'

// Based on Unlit shader, but culls the
front faces instead of the back

Shader "Inside out" {
Properties {
    _MainTex ("Base (RGB)", 2D) =
    "white" {}
}

SubShader {
    Tags { "RenderType"="Opaque" }
    Cull front // ADDED BY BERNIE, TO
    FLIP THE SURFACES
    LOD 100

    Pass {
        CGPROGRAM
        #pragma vertex vert
        #pragma fragment frag

        #include "Unity_CG . cginc"
        struct app_data_t {
            float4 vertex :
            POSITION;
            float2 tex coord :
            TEXCOORD0;
        };

        struct v2f {
            float4 vertex :

```

图 5 unity 内部脚本代码部分示意图

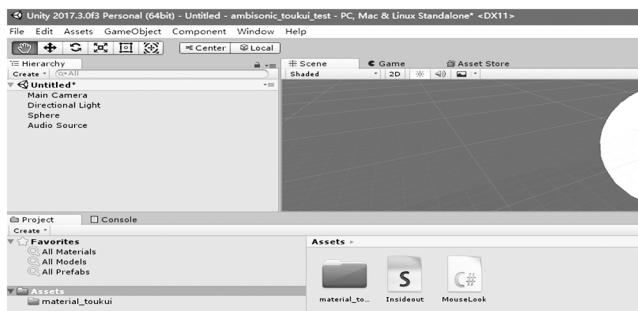


图 6 图 5 unity 内部界面部分截图，带有 C# 和 S 字样的图标即为本次使用的代码脚本文件

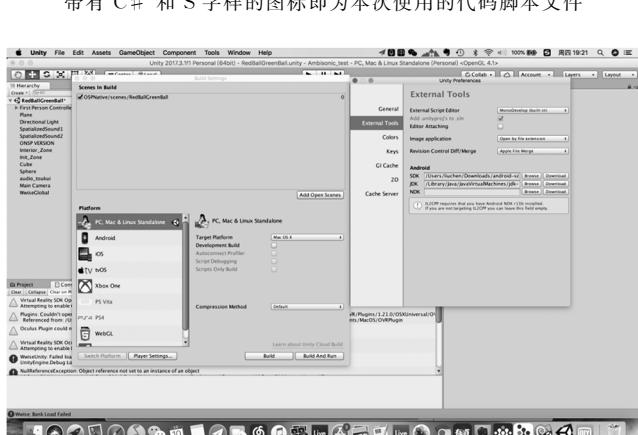


图 7 unity 导出设置，可根据需要进行多平台输出

Unity 的导出菜单中可选择需要的播放平台，

点击 build 将工程导出为一个可运行的应用程序。由于实时引擎的相对开源，通常支持多平台的发布。以 Unity 为例，游戏引擎内制作好的内容可以打包至本地运行，也可对应需求的操作系统（如 Windows、MacOS、Android）进行导出，实现全景视频的多平台播放。

五、全景视频播放系统中的声音

audio source 是 Unity 主要的音频播放组件，在实时引擎中可通过添加多个 audio source 同时播放、处理多条的音频文件（即“多轨播放”），对不同的音频需求进行处理，甚至可以简单添加混响、3D 衰减等功能。由于 video player 自带的音频播放端口不支持立体声以外的音频文件，因此在处理其他格式的音频时（例如 Ambisonic），需要将音频挂在 audio source 后指定到 video player 的关联接口上。

此外，Unity 目前并不支持对 Ambisonic 的编解码，而要依赖 Google、Microsoft 等大厂提供的 SDK（Software Development ToolKit，软件开发工具包）进行编解码操作和播放。解码质量和算法由各平台的 SDK 来设置。在 project settings-Audio 菜单的 audio manager 选项卡中把 spatializer plugin 和 Ambisonic decode plugin 按照当前使用的 SDK 需求进行选择。

除了在 Unity 内部进行声音的配置，还可以通过集成第三方插件的方式来实现多样化的音频播放处理。以加拿大 AudioKinetic 公司开发的音频中间件 wwise 为例，作为一款被广泛地应用于国内外互动音频开发领域的音频引擎，声音设计师可以更加自由地设计声音播放规则并在开发环境中进行效果的迭代。

例如将游戏场景中声源作为“对象”（object）实现基于距离、空间的实时声处理；复杂播放规则的实现与控制；通过自定义游戏参数来管理复杂的声音设计效果等。Wwise 使用事件（Event）作为引擎中播放声音的基础单元。在集成到 Unity 后通过调用“事件”来实现设计后的音频播放规则。这样做的好处是在制作需求比较复杂的作品中，只需调用很少量的代码即可得到想要的效果。近两年

wwise 增加了 Auro3D、Reflect 等插件的支持，极大地丰富了声音设计师在空间音频（spatial audio）领域中的技术手段。无论是 Ambisonics 或是基于射线的实时混响的技术，可以说在塑造沉浸式音频上，Wwise 有着令人惊喜的表现。

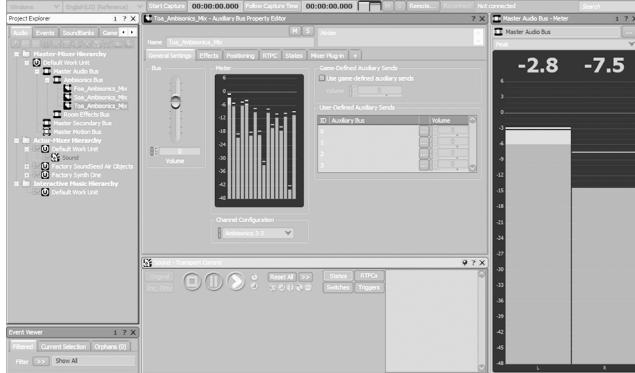


图 8 频中间件 wwise 处理 ambisonic 格式音频示意图

六、结语

基于实时引擎的还放系统固然解决了一些问题，比如在 Ambisonics 的播放和编解码，但缺少有效的时间码同步系统，测试时产生了一定的播放延迟、声画不同步的问题。这有待从底层的角度进行更加有效的音频接入和插件的开发。遗憾的是本文在撰写过程中暂时未能找到更好的解决方案。

阅读官方文档得知，各个平台的相关 SDK 进行编解码运算时会产生较高的 CPU 占用率，笔者限于自身条件，在测试中无法对流畅播放全景音视频所需的系统配置要求进行精确测量，因此无法得出较为明确的结论。

全景视频在今天能够作为一种新的影像表现形式，对影视从业者提出新的问题和挑战，这无疑是有利于这个行业的进步的。同时我们也应当及时梳理技术的发展脉络，拓宽研究视野，为新的影像表达发掘更具有活力的武器。◆

【作者信息】刘辰，男，山东青岛人，北京电影学院 2018 级电影声音制作与理论研究生，北京 100088

参考文献

- [1] 崔蕴鹏. 巨幕、立体到 VR：影像“距离”空间的补全 [J]. 当代电影.
- [2] Oculus Audio SDK: Manual Documentation.
- [3] Wikipedia: Ambisonic Data Change Formats.
- [4] Microsoft Resonance SDK: Documentation.
- [5] Unity3D: Manual Documentation.
- [6] 王维. 我国第一座大型球幕电影厅 [J]. 今日电子, 1994, (10).