

沉浸式全景视频技术指南

Immersive 360 Degree Video Technology Guide



媒体技术实验室，上海交通大学

Media Lab, SJTU

2018. 8

前言

《沉浸式全景视频技术指南》由上海交通大学媒体技术实验室编著、面向虚拟现实视频领域的研发人员、技术爱好者以及相关专业的本科生、研究生。我们从多媒体系统的角度，根据虚拟现实全景视频服务端到端系统涉及的主要技术，组织了 6 个章节，进行整体介绍和梳理，力求做到全面、主流、新颖。具体目录如下：

- 第一章 全景视频生成技术
- 第二章 全景视频呈现技术
- 第三章 全景视频处理技术
- 第四章 全景视频流媒体技术
- 第五章 全景视频 QoE 技术
- 第六章 相关国际标准组织

秉承开、共享、进化的精神，我们将定期更新最新稿到 Github 上：

<https://github.com/sjtu-medialab/ImmersiveVideoTech>

关于虚拟现实的基础知识，本书中没有过多介绍，建议读者参阅我们翻译的《Virtual Reality》中文译版，这本书的英文版本由 UIUC 的 Steven M. LaValle 教授 2016 年网上免费公开，内容全而新。中文译本可以从这里获得：

<https://github.com/sjtu-medialab/VirtualReality-Chinese>

我们欢迎来自读者任何问题、批评、建议，以期不断完善更新，可以直接在 Github 中提出 issue，或通过邮箱与我们联系：sjtu_medialab@163.com

注：版权归作者所有，未经授权禁止转载、摘编、复制或建立镜像，禁止未经授权的商业使用行为。

目 录

第一章 全景视频生成技术.....	4
1. 1 成像原理及技术.....	4
1. 2 全景图像拼接技术.....	15
1. 3 开源软件.....	25
1. 4 典型全景相机介绍.....	27
第二章 全景视频呈现技术.....	33
2. 1 HMD 的基本组成及典型设备.....	33
2. 2 渲染相关技术.....	40
2. 3 光学相关技术（反畸变）.....	49
2. 4 常用的终端集成引擎.....	57
2. 5 音频相关技术.....	64
2. 6 数据转换技术.....	72
第三章 全景视频处理技术.....	79
3. 1 全景图编辑处理软件.....	79
3. 2 全景非线性编辑软件.....	81
3. 3 视频稳像技术.....	89
第四章 全景视频流媒体技术.....	97
4. 1 MPEG OMAF.....	97
4. 2 HEVC, DASH 等相关扩展.....	108
4. 3 流媒体系统.....	112
4. 4 沉浸式流媒体优化技术.....	129
第五章 全景视频 QoE 技术.....	161
5. 1 主客观评价简介及主观评价的实施案例.....	161
5. 2 典型的 QoE 要素.....	170
5. 3 球面指标测度.....	175
第六章 相关国际标准组织.....	186
6. 1 MPEG-I.....	187
6. 2 IEEE P2048.....	192
6. 3 IEEE P3333. 3.....	193
6. 4 Khronos.....	194
6. 5 WebVR.....	195
6. 6 3GPP.....	196
6. 7 DVB.....	197
6. 8 VRIF.....	198
6. 9 QoE: QUALINET, VQEG, ITU-T.....	199
6. 10 DASH-IF.....	208
6. 11 CTA.....	208
6. 12 SMPTE.....	208
6. 13 ETSI.....	209
6. 14 SVA.....	209

第一章 全景视频生成技术

“全景”的概念最早是在欧洲艺术领域提出的，随后工程师们渐渐发现“全景”的优点和发展前景，全景应用由艺术领域转向了工程。全景视频，也称为360度视频，在近年来随着虚拟现实（VR）技术的快速发展而兴起，成为未来视频服务的新型载体。全景视频因其以用户所在的虚拟位置为球心，展示出整个球面场景而得名，其通过专业的360°全景摄像机拍摄，并利用相关技术以及软件处理后生成。更进一步而言，沉浸式的全景视频，即目前通过头戴式显示器（HMD）观看的全景视频，还需依靠立体成像技术方可具备良好的立体感和沉浸感，让用户感受到360°无死角的视频体验。

1.1 成像原理及技术

1.1.1 全景成像原理及技术

全景立体成像技术可为机器人导航、军事侦查、计算机视觉、虚拟现实等领域提供大视角场景的立体感知和重现功能，近年来发展快速，成为光电子学、计算机视觉和计算机图形学的研究热点。全景成像（panoramic imaging, PI）是指利用特殊的成像装置从一个视点获取水平方向一周360°、垂直方向大到半球以上视场的多方向成像；而全景立体成像技术则是基于全景成像技术，利用双目视觉原理获取全景立体信息，该立体图像对的视差反映了场景的深度信息。

本节将就目前广泛应用的几种全景立体成像方法进行原理技术及应用的介绍。

全景立体成像特性

全景成像是在一个视点对周围所有方向的场景成像，而立体成像需要两个或多个视点，在垂直于两视点连线的方向上获得的场景具有立体感，但在两视点连线方向上不会具有立体感。

全景成像基本方法

全景成像系统一般由全景视觉光学成像系统、图像传感器、图像处理系统和输出图像显示器等部分组成。目前，全景成像的系统或方法主要有以下几种：

（1）图像拼接成像

该方法使相机围绕其光心的垂直轴线旋转一周，进而对水平一周多个不同方向的场景成像，再将这些不同方向的场景图像进行拼接，获得一幅全景图像。因而，其中的硬件部分必须有精确的转动机构。图1.1所示的是应用该方法的旋转式云台相机。



图 1.1 云台相机

旋转式全景相机问世较早，1844 年德国人已经试验成功全景摇头转机。1920 年，Stromberg 等人发明了旋转式 360° 全景相机。1922 年，Richards 设计可以调整镜头和转轴的位置、底片放置方式的方案，研制出高质量旋转式全景相机。旋转式全景相机在中国使用的历史也比较长，主要是照相馆用其拍摄团体照，我国的上海、天津等地的照相机厂都曾生产过，至今仍有新的品种，但以胶片相机为主。目前国外已有彩色线阵 CCD 旋转式数码相机，在进口的同类相机中，瑞士生产的“环摄”转机最为著名，由瑞士 Seitz 公司生产的一款像素过亿的全景数码相机，像素可达 4 亿以上。这款 360 度全景相机每次拍摄大概要花费 2s 的时间，采用 80mm 的镜头时像素可达 4.7 亿。此相机拍摄相片的分辨率可达 7500x21250 像素。2004 年 1 月美国“勇气”号火星探测器登陆火星，上面安装有旋转式全景视觉系统。美国宇航局公布的“勇气”号火星探测器拍摄的火星三维全景黑白照及火星表面高分辨率全景彩照，都是由旋转式全景视觉系统获得。

该方法虽然成像分辨率高，但成像速度较慢，拼接算法复杂，不满足单一视点约束，一般只能拼接出圆柱面全景图像，而最大的不足是不能实时动态全景成像。

另一种基于图像拼接的方法是利用面向不同方向的多个相机同步拍摄多幅图像，进而将同步采集的多幅图像进行融合拼接得到全景图像。为获取水平方向 360° 和垂直方向一定角度的场景信息，根据不同方位镜头视场的大小确定所需相机个数。如要求该成像设备满足单一视点要求，则各台相机的光学中心必须重合。目前，在全景图像应用领域比较常见的就是这种多相机拼接式全景视觉系统，这类系统在军民领域均已得到了广泛的应用。

由此延伸的关于全景图像拼接方面的具体技术还将在后续章节中详细介绍。

(2) 鱼眼镜头成像

鱼眼镜头（图 1.2）是一种极端的超广角物镜，其焦距非常短（小于 16mm），能获取接近或大于 180° 的全景视场。鱼眼镜头由于获得大视场角，存在较大的桶形畸变，除了图像中心的景物保持不变，其它本应水平和垂直的景物沿各个方向从中心向外辐射，发生较大的畸变。



图 1.2 鱼眼镜头

鱼眼镜头的发展经历了从水下鱼眼的简单模仿到目前 270° 超大视场鱼眼镜头的过程。1919 年，Wood 在一个装满水的容器上盖一块玻璃板，构成鱼眼相机来实现超广角摄影，这套装置是对水下鱼眼最简单的模仿。1922 年 Bood 改进了 Wood 的装置，用半球玻璃透镜取代装满水的容器构成全景成像装置。1923 年，Hill 改进 Bood 的设计，在半球透镜前面引入一个光焦度绝对值很大的负弯月形透镜，改进后的镜头能够拍到较好的半球空域的云层照片，所以这种结构也叫做希尔天空镜头，之后的鱼眼镜头设计都沿用这种设计思想。后来一些学者继续对鱼眼镜头做了改进，提升了鱼眼镜头的成像质量。20 世纪 60 年代，光学自动设计技术的应用，使鱼眼镜头的发展更加迅速，出现了很多像质更加优良的光学结构。1964

年, Miyamoto 设计的镜头, $2\omega = 180^\circ$, $f = 16mm$, 不仅使系统像差得到了较好的校正, 并且像面照度的均匀性有了明显的改善。有些鱼眼镜头视场角能达到 220° , 有的甚至能达到 270° 。鱼眼镜头在摄影、医疗和安全监测等领域都有一定的应用。

高质量的鱼眼镜头通常需要采用 10 片以上的高质量光学材料结构, 因此还具有系统复杂、造价成本昂贵等缺点。

(3) 反射式成像

反射式全景视觉系统是由一台摄像机和一个反射曲面构成, 它是把反射曲面和传统的透射成像透镜组合在一起, 实现大视场成像的一种光学结构, 全景成像投影原理与鱼眼镜头全景成像的投影原理类似。摄像机垂直放置, 光轴垂直于水平面, 在摄像机上面放置一个曲面反射镜, 反射曲面将与光轴夹角很大的入射光线反射成为夹角较小的光线, 然后经摄像机成像, 从而增大视场。现有的应用于反射式全景视觉系统的反射面有抛物面、椭球面、双曲面等。根据反射面不同, 反射式全景视觉系统又可以分为抛物面反射式全景视觉系统、椭球面反射式全景视觉系统、双曲面反射式全景视觉系统等。

自从 1970 年美国宾夕法尼亚大学 W. Rees 设计了一套双曲反射镜面全景成像系统并成功应用于塔楼士兵监测周围目标后, 各国科研人员对反射式全景视觉系统进一步拓展。Yamazawa 等人利用双曲面反射镜与一个透镜相结合构成双曲面反射式全景成像系统。Chah1 等人在前人研究基础之上, 具体给出了双曲面反射镜的曲面公式。A. Gardel 等人则开展了反射式全景嵌入式系统的研究。美国 DARPA 将反射式全景视觉系统用于 Urbie 战术侦察机器人, 主要用于城市地形战术侦察。

国内反射式全景视觉系统的研究工作起步于 20 世纪 90 年代中期, 四川大学曾吉勇博士和苏显渝教授对抛物面反射镜、双曲面反射镜等全景视觉系统的原理、结构、成像特点和展开效果做了比较系统的研究。哈尔滨工程大学朱齐丹教授开展了高分辨率嵌入式反射全景视频处理系统的研究, 全景图像的分辨率为 2048×2048 。北京理工大学李科杰、张振海、王健等人开展了高分辨率高灵敏度抛物面反射式全景视觉系统研究, 作用距离约 $1.5km$, 有效全景图像的分辨率为 2750×2200 , 605 万像素, 帧率接近实时 25 帧, 最低照度可达 $0.0003lx$ 。

(4) 折反射式成像

折反射成像利用镜头前的反射镜扩大相机的视场从而获取实时全方位图像。这种成像方法, 能够实现实时大视场成像, 结构和几何计算简单, 立体图像对应点匹配容易等优点, 且成本较低。

折反射全景成像基本原理

折反射全景成像是利用曲面反射镜把水平方向 360° 范围内物体的光线反射到成像传感器, 从而一次性拍摄获得远大于普通相机视场范围的景象。与图像拼接和鱼眼镜头技术相比, 折反射全景成像技术能一次性实时获取 360° 图像, 具有大于半球空间的视场、成像装置设计简单、成本低等优点。折反射全景成像系统主要由三部分组成: 感光元件 (CCD 或 CMOS 器件)、成像透镜 (如常规成像透镜或远心透镜) 和反射镜, 如图 1.3 所示。由折反射全景成像系统获取的原始图像 (即球形图) 称为折反射全景图像, 如图 1.4 的上图所示。但由于原始图像存在严重变形, 不适合直接观察, 需要把折反射全景图像变换为符合人眼视觉习惯的圆柱面投影图像, 如图 1.4 的下图所示。根据是否满足单视点约束, 折反射全景成像系统分为单视点成像系统和非单视点成像系统。单视点成像的优势在于能将全景图转换成无畸变的平面透视图, 且能保证成像系统使用一台相机一次捕获到所需要的信息, 观察视角更大, 一般的图像处理方法可以用来进行图像分析和处理。

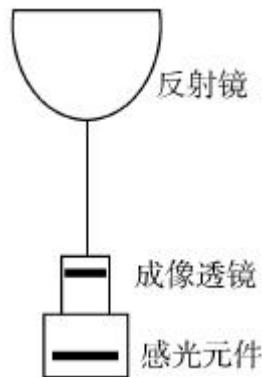


图 1.3 折反射全景成像系统组成

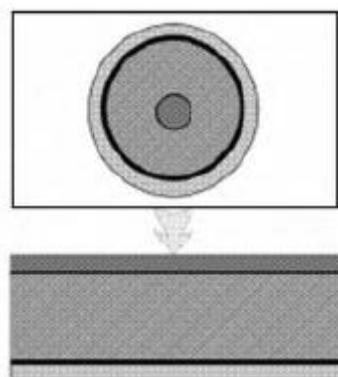


图 1.4 全景图像展开为柱面图像示意图

折反射全景成像反射镜面形式

折反射全景成像系统需要一个经过特殊加工的表面光滑的反射镜，将周围空间的光线汇集到反射镜上（分为凸面反射和凹面反射）。反射镜根据镜面线型的不同，又分为球面、双曲面、抛物面、圆锥面、椭球面等类型，如图 1.5 所示。

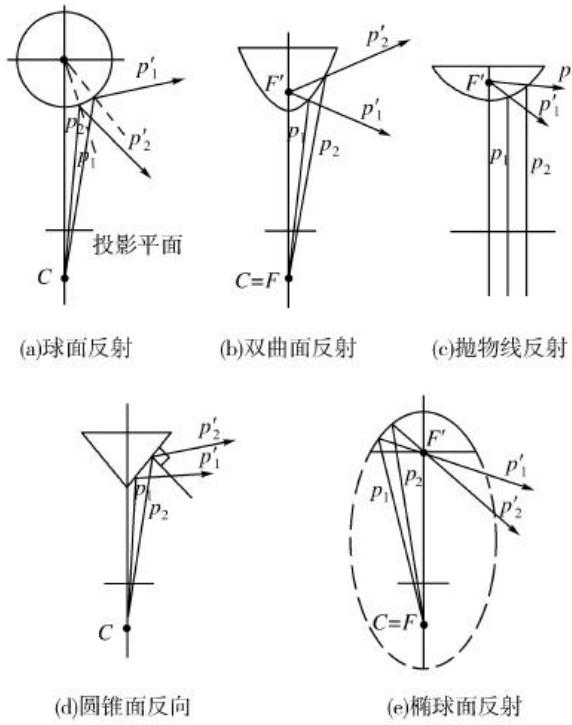


图 1.5 反射镜光路示意图

折反射成像映射原理

单视点全景成像系统，其所有入射光线的延长线都相交于同一视点，根据光路跟踪原理，对成像点进行光路跟踪和投影变换，可以建立单视点全景图像到柱面全景图像像素点间的坐标映射关系。如图 1.6 所示。把图 1.6(a) 中的虚拟圆柱面沿与 x 轴正方向相交的一条母线展开，并以其左下角顶点为原点建立二维直角坐标系，可得图 1.6(c) 所示的柱面全景图像，图 1.6(b) 表示全景图像。根据不同的反射镜面类型和参数，推算的像素坐标映射关系也不同。另外，成像系统焦距、位置以及分辨率对像素坐标映射关系也有影响。

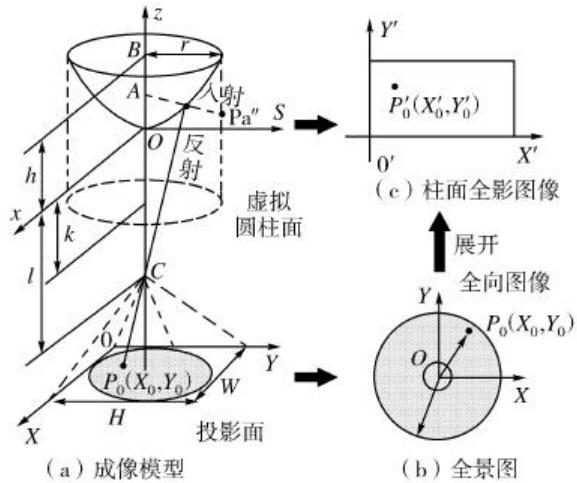


图 1.6 全景图转换为柱面展开图

折反射全景立体成像系统

利用 2 至 3 个折反射全景成像系统就可以实现全景立体成像。目前，使用两组折反射全景成像系统构成的立体成像系统装置结构主要有三种：(1) 水平方向上并列放置两组独立的

全景成像装置，两组装置的轴线均垂直于水平方向，如图 1.7(a)所示。系统结构简单，但设备间会有一定相互遮挡，进而影响该遮挡区域真实场景的成像；(2)垂直方向上同轴上下放置两组独立的全景成像设备，如图 1.7(b)所示。虽然避免了第一种系统之间的相互遮挡，且立体图像对间存在良好的极线约束。但由于使用了两个成像设备，采集的立体图像对会引起曝光、颜色等方面差别的；(3)与第二种不同的是上下两组全景成像设备共用同一个相机，如图 1.7(c)所示。该系统要求相机拍摄到的一个反射面中要有另一反射面的图像，即在一幅图像中同时获取两个全景图像，避免了前两种系统使用两个相机两次成像导致的缺陷，但成像分辨率不高。

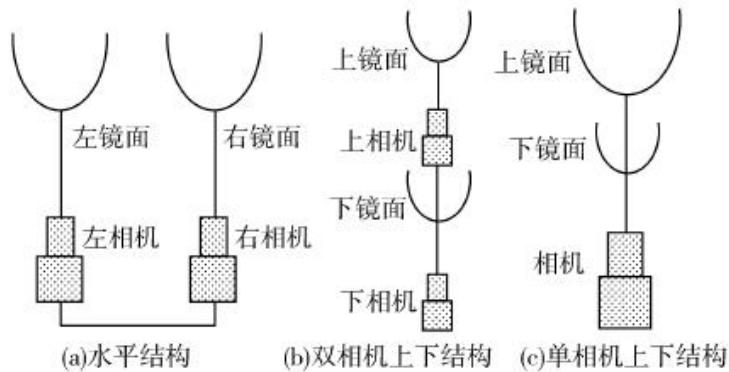


图 1.7 三种不同结构的折反射全景立体成像装置

使用三个折反射全景成像设备构成的全景立体成像系统装置结构如图 1.8 所示。三个折反射全景成像系统 A、B、C 分别放置在等边三角形的三个顶点上，每个全景成像系统水平方向的视场都是 360° ，均可等分为 I、II、III 三个区域。全景立体图像对取自 A、B、C 三个系统不同的视场区域。视场区域 A_1 、 B_1 、 C_1 和 A_{II} 、 B_{II} 、 C_{II} 分别组成了完整的左右眼全景图像，而 A_{III} 、 B_{III} 、 C_{III} 为相互遮挡的区域。该全景立体成像数据处理相对简单，能实现对动态景物的全景立体成像，但硬件结构较复杂，成本较高。

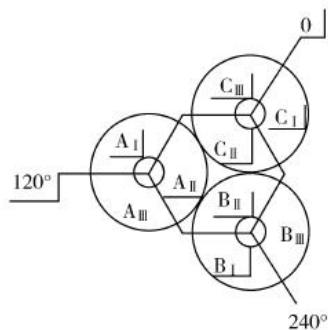


图 1.8 全景立体成像系统

Greguss 教授首先提出 PAL 全景透镜的结构，并进一步提出使用抛物面设计 PAL 的两个透射面和两个反射面，得到更精准的成像结果，并尝试设计全景环带透镜光学系统。进入 90 年代，Powell 对全景环带光学系统的结构开展深入研究，首次提出全球面 PAL 结构，与此同时，美国阿拉巴马大学 Gilbert 教授、Fair 教授与马奎特大学 Matthys 教授等人对 PAL 光学系统结构开展仿真计算，并对 PAL 光学系统的应用做了深入探讨，并详细讨论了物面测绘定位和环带像面展开的方法。索尼公司研制 360° 折反式全景相机分辨率约为 38 万像素，通过 DSP 全景图像算法程序，最终输出图像输出 7.5 帧 CCD 全景图像。这种新型的全景摄影装置主要用于监控领域。

国内开展折反式全景视觉系统研究的单位主要有浙江大学、北京理工大学等。浙江大学

开展了较长焦距全景环带成像光学系统研究，设计的小口径 PAL 结构的视场角为 50° 全景成像光学镜头。北京理工大学李科杰、张振海首次研制折反射式微光全景视觉系统，成像视场大小为水平方向 360°，垂直方向仰角 37°，俯角 18°，最低照度可达 0.0003lx，可实时输出 PAL 标准制式微光全景图像，通过研制 FPGA+DSP 嵌入式全景图像系统，将全景环形图像展开、畸变校正为 8 个方向全景图像，可用于地面侦察机器人微光条件下的全景图像信息获取。

小结

全景成像技术由于其视场范围大的优点，广泛应用于机器人导航、军事侦查、计算机视觉、虚拟现实等领域。本节介绍了几种主要的全景立体成像技术，包括旋转图像拼接、鱼眼镜头、反射和折反射成像四种方法及其各自优缺点。用于生成沉浸式全景视频的典型和新型全景相机/系统将在 1.4 节中进行详细介绍。

1.1.2 3D 成像原理及技术

现实世界是一个三维立体世界，随着社会的发展，传统的二维平面显示在某些方面已不能满足人类的需求，人们希望显示器或其他设备能真实地还原显示出空间的三维信息。因此，三维立体显示应运而生，并不断得到发展，一度成为显示领域的一个研究热点。

19 世纪 30 年代，科学家 Wheatstone 着手研究人的视觉，并于 1838 年发明了立体镜，拉开了人类对三维立体显示研究的帷幕。20 世纪中后期，随着计算机信息领域的发展以及平板显示器的出现，三维立体显示蓬勃发展。日本、韩国、欧美等国家从 20 世纪 80 年代开始了三维立体显示的基础研究，各国根据自身的情况，开发了各种技术和产品。我国三维立体显示虽起步较晚，基础薄弱，与国际水平存在差距，但也有不少高校、研究所和公司企业对立体显示开展研究并取得了一些成果。

三维立体显示目前被广泛应用，而对于需要立体感、沉浸感的全景视频而言，该技术同样关键。如果说黑白显示器是第一代产品，彩色显示器是第二代产品，那么立体显示器就是第三代产品。自 19 世纪 30 年代开始，经过接近两个世纪的发展，已开发出各种三维立体显示。本节按其基本工作原理是否为双目视差将三维立体显示分为两大类。所谓双目视差是指人两眼间有一定瞳距，大约有 6~7 厘米的间隔，在观看物体时左眼和右眼所接收到的视觉图像略有差异。基于双目视差原理的三维立体显示为观看者的左右眼提供同一场景的立体图像对，采用光学等手段让观看者的左右眼分别只看到对应的左右眼图像，这样便使观看者感知到立体图像。这类三维立体显示的技术相对成熟并有相应产品，但存在观看视疲劳等问题。非基于双目视差原理的各种三维立体显示的工作原理各不相同，如利用光学干涉衍射原理、人眼视觉暂留效应以及人眼视错原理等。这类三维立体显示不存在观看视疲劳，但技术相对而言尚未成熟。下面对这些三维立体显示技术作简要阐述。

基于双目视差原理的三维立体显示

早在 19 世纪摄影技术刚刚起步时，人们就用两台性能和参数完全相同的相机并列，模拟人的左右双眼，同时拍下两张有微小差异的相片，之后再透过平行视线法、交叉视线法或类似双筒望远镜的专属观看设备等，让人的左右双眼分别观看两张并列拍摄的相片，以重现视差，借以模拟出立体视觉。

目前，基于双目视差原理的三维立体显示技术主要有眼镜/头盔式立体显示和光栅式自由立体显示两类，前者技术和产品都较为成熟，后者已有一些产品，但其性能大多有待提高。

(1) 眼镜/头盔式三维立体显示

眼镜三维立体显示按其工作原理主要分为三类：

一是基于波长，如红绿、红蓝等互补色立体眼镜的三维立体显示，其利用人类能感知红蓝绿三原色的机理，使进入左右眼的光谱不同，从而形成双眼视差。以红蓝光眼镜为例，片源是由红、蓝光染色生成的具有双目视差的立体图像，由于相同颜色镜片会过滤掉相同颜色的图像。因此，通过红色镜片观察的眼睛，只能看到蓝色图像，同样通过蓝色镜片观察的眼睛，只能看到红色图像，由于红蓝光产生的图像本身具有差异，从而产生立体视觉。其主要特点是技术成熟、成本低，但不能显示彩色图像。



图 1.9 红蓝 3D 眼镜

二是基于时序立体眼镜的三维立体显示，其显示屏分时显示左右视差图，并通过同步信号发射器及同步信号接收器控制观看者所佩戴的液晶快门立体眼镜，使得当显示屏显示左(右)眼视差图像时左(右)眼镜片透光而右(左)眼镜片不透光，按照上述方法将两套画面以极快的速度切换，在人眼视觉暂留特性的作用下就合成了连续的画面。其主要特点是要求显示器的帧频为普通显示器的两倍，一般需要达到 120 Hz。

三是基于偏振眼镜的三维立体显示，显示屏上左右眼视差图的光线为互相正交的线偏振光或左右旋圆偏振光，通过偏光滤镜或偏光片滤除特定角度偏振光以外的所有光，让 0 度的偏振光只进入右眼，90 度偏振光只进入左眼，即双眼分别看到立体图像对中不同的图像，因而观众配戴上相对应的偏光眼镜就可以观看到立体效果。



图 1.10 偏振式 3D 眼镜

头盔式三维立体显示是在观看者双眼前各放置一个显示屏，观看者的左右眼只能分别观看到显示在对应屏上的左右视差图，同时遮挡住外部的光线和视野，从而提供给观看者一种沉浸于虚拟世界的沉浸感。显示器一般与电脑或手机的视频输出连接，分别提供给左右眼不同的图像。头盔显示器提供一个稳定的双眼视差三维影像，利用附带的头部跟踪器后可实时获取头部的位置和运动信息传递给生成图像的电脑或手机，这样就可以根据观察者的位置和方位信息提供不同图像画面。目前，全景视频的成像、显示、播放主要通过 HMD 以及相关设备完成。这种立体显示存在单用户性、显示屏分辨率低、头盔沉重及易给用户带来不适感等缺点。



图 1.11 用于观看沉浸式全景视频的 HMD

(2) 光栅式三维自由立体显示

光栅式自由立体显示器主要是由平板显示屏和光栅精密组合而成，左右眼视差图像按一定规律排列并显示在平板显示屏上，然后利用光栅的分光作用将左右眼视差图像的光线向不同方向传播，当观看者位于合适的观看区域时其左右眼分别观看到左右眼视差图像，经过大脑融合便可观看到有立体感的图像。根据采用的光栅类型可分为狭缝光栅式自由立体显示和柱透镜光栅式自由立体显示两类。

狭缝光栅式自由立体显示器又分为前置狭缝光栅和后置狭缝光栅两种，其结构与原理图分别如图 1.12(a) 和图 1.12(b) 所示。图 1.12(a) 中狭缝光栅置于平板显示屏与观看者之间，观看者左右眼透过狭缝光栅的透光部分只能看到对应的左右眼视差图像，由此产生立体视觉。图 1.12(b) 中狭缝光栅置于平板显示屏与背光源之间，用来将背光源调制成狭缝光源。当观看者位于合适的观看区域时，从左(右)眼处看显示屏上只有左(右)眼视差图像被狭缝光源照亮，那么左右眼就只能看到对应的左右眼视差图像，由此产生立体视觉。

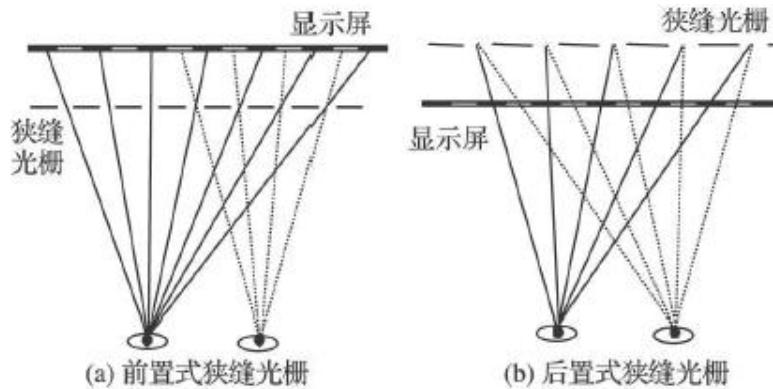


图 1.12 狹缝光栅式自由立体显示器的结构与原理

柱透镜光栅式自由立体显示器的结构与原理如图 1.13 所示，利用柱透镜阵列对光线的折射作用，将左右眼视差图像分别提供给观看者的左右眼，经过大脑融合后产生具有纵深感的立体图像。

狭缝光栅式自由立体显示由于狭缝不透光部分对光线的遮挡，从而导致立体图像亮度相对于平面图像损失严重；而柱透镜光栅式自由立体显示由于采用透明的柱透镜，除了透镜对光线的吸收外立体图像亮度基本没有损失。这两种光栅式立体显示由于在平板显示器上同时显示两幅或多幅视差图像，从而导致立体图像分辨率相对于平面图像有所降低。光栅式自由立体显示由于其结构简单、易于实现、无需佩戴辅助观看装置及立体显示效果良好等优点得到立体显示研究人员的青睐。然而要使光栅式自由立体显示得到广泛应用，还需要研究人员不断提高其性能，改善其立体显示效果。

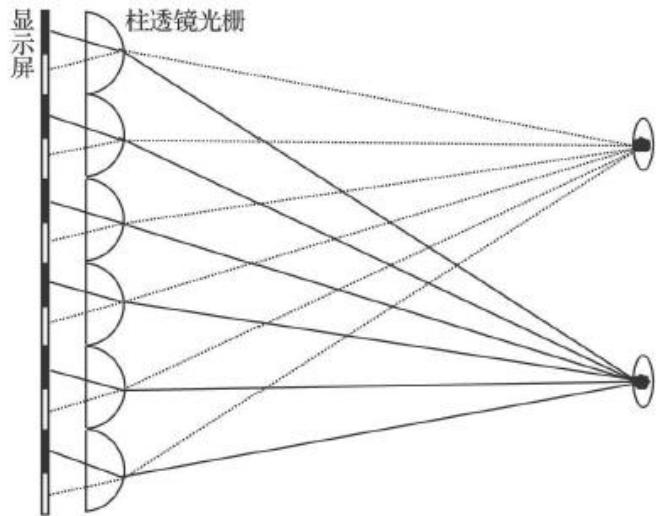


图 1.13 柱透镜光栅式自由立体显示器的结构与原理

(3) 自动分光立体显示

近年来，不需要戴眼镜或头盔的自动分光立体显示器已经从实验室进入市场，大多数自动分光立体显示器是基于平板显示器的系统，通过结合水平的双眼视差和运动视差给出了不戴眼镜的 3D 立体效果。人们在真实的世界里观察主要有两种视觉感受，一是双眼会看到不同画面的双眼视差，二是头部处于不同位置或运动都会看到不同的画面，相当于沉浸在一组无数张画面组成的场景。自动分光立体显示技术便利用到上述的视觉特性，通过将原视景划分为几个视景组，使得人双眼落在不同的视景组中，左右便会看到不同画面，从而产生双眼视差。同样，处于不同位置或运动，人眼视线也同时落在不同视景组里，便能实现水平运动视差。该技术特点是不需要头部跟踪以及相关的复杂技术，多个观察者可同时在观察区域中自由地移动双眼，获取多视角的 3D 效果。然而，建立自动分光系统比较困难，同时产生所有视图可能使观察者看见错误的图像，且偏离理想的距离越远，这种可能性更大。

非基于双目视差原理的三维立体显示

非基于双目视差原理的三维立体显示主要有全息立体显示、集成成像立体显示以及体显示等，它们的技术和产品尚不成熟，需要开展更深入的研究工作。

(1) 全息立体显示

全息技术是利用干涉原理将物体发出的特定光波以干涉条纹的形式记录下来，形成“全息图”，全息图中包含了物光波前的振幅及相位信息。当用相干光源照射全息图时，基于衍射原理重现原始物光波，从而形成原物体逼真三维图像。全息立体显示是一种真三维立体显示技术，观看全息立体图像时具有观看真实物体一样的立体感。全息图的每一部分都记录了物体各点的光信息，故即使全息图有所损坏也照样能再现原物体的整个图像。通过多次曝光可在同一张底片上记录多个不同图像且互不干扰地分别显示出来。

近年来，随着计算机技术的发展和高分辨率电荷耦合成像器件(Charge Couple Device, CCD)的出现，数字全息技术得到迅速发展。与传统全息不同的是，数字全息用 CCD 代替普通全息记录材料记录全息图，用计算机模拟取代光学衍射来实现物体再现，实现了全息图记录、存储、处理和再现全过程的数字化。数字全息技术和显示技术的结合使全息技术的应用前景更加广阔，为数字全息在真 3D 显示中的应用奠定了良好的基础。目前，数字全息技术主要有三种：基于数字合成全息技术的三维显示技术、基于空间光调制器的数字全息三维显

示技术、基于集成技术的数字全息三维显示技术。

(2) 集成成像立体显示

集成成像技术也由记录和再现两个基本过程组成，与全息技术不同的是其记录再现过程并不需要相干光的参与，而是利用二维微透镜阵列来实现。如图 1.14(a)所示，第一步将记录介质置于微透镜板的焦平面上，每个微透镜单元从不同的方向记录物体空间的部分信息，将每个微透镜单元所记录的一幅幅小图像称为“子图像”，空间物体的视差信息就被这一幅幅“子图像”记录下来。第二步将记录介质置于具有相同参数的微透镜阵列的焦平面上，用散射光照射，由光线可逆原理就可在微透镜板另一侧观看到再现物体空间场景。如图 1.14(b)所示。

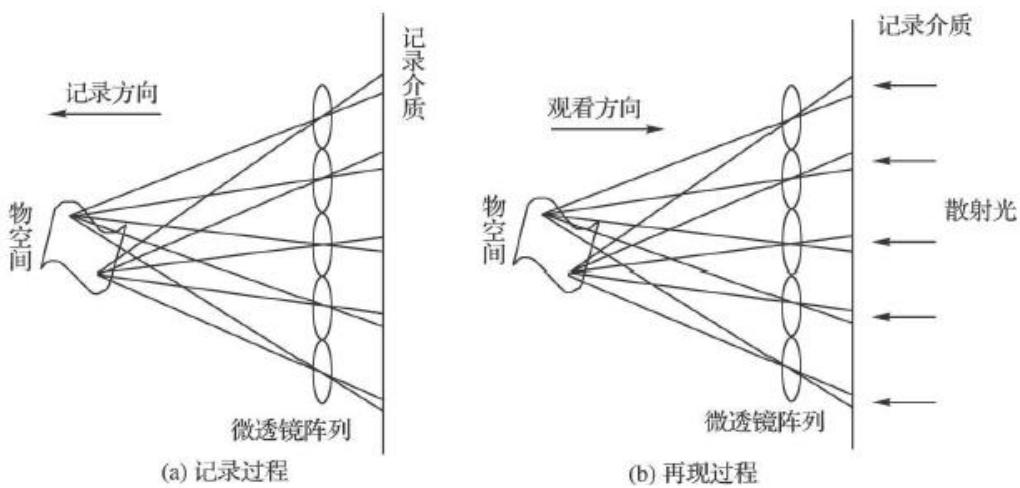


图 1.14 集成成像技术的记录和再现过程

利用集成成像技术实现三维立体显示可供多个观看者同时观看且无需配戴特殊眼镜，观看三维图像时不存在眼睛会聚与调节不匹配的问题，再现的三维场景具有全真色彩以及连续视差。但集成成像技术也具有其自身的不足之处需要改善：1) 记录和再现两过程中会产生空间反转，通过两次记录可解决空间反转问题，但图像质量会有所下降，采用负透镜阵列等技术可较好地解决空间反转问题；2) 立体观看视角比较窄，可通过透镜开关及非球面透镜等方法来拓宽立体视角；3) 可清晰记录的物体景深比较小，一般为几个厘米，可利用双图像平面等方法来增强景深；4) 分辨率低，可采用逆光线获取等方法来提高分辨率。此外，子图像间的相互串扰以及透镜单元本身的像差等因素也会影响到集成成像质量。随着对集成成像理论研究的不断深入以及对成像质量的不断改善，集成成像技术将成为立体显示领域的重要研究方向。

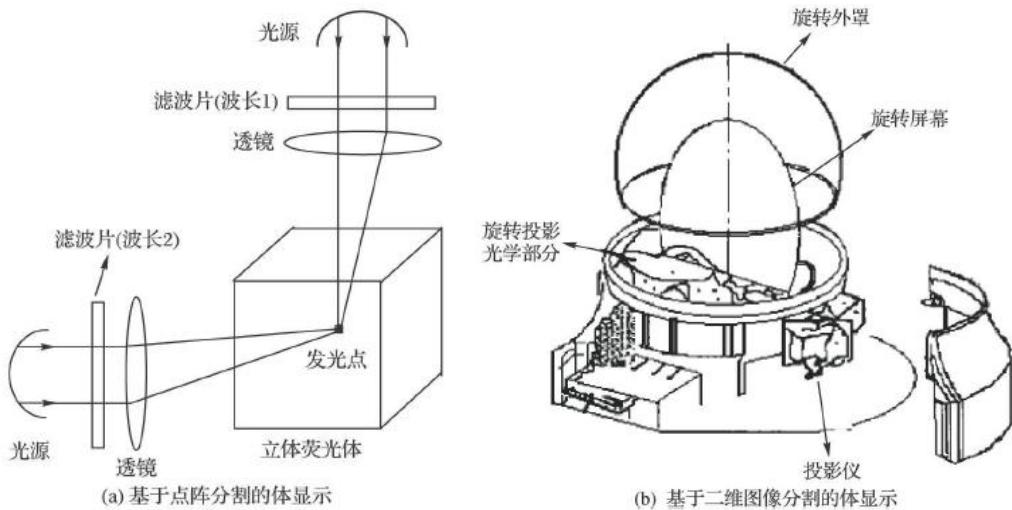


图 1.15 体显示的工作原理

(3) 体显示

近年来，随着计算机数据处理能力的迅速提高和数据存储技术的发展，采用嵌入式系统的3D立体显示器引起人们的广泛关注。这种显示器利用屏幕的旋转和光投影等技术，将原来的二维图像切片合成富有真实立体感的三维图像，和真实物体的视觉效果比较接近。可以让多人从多角度观察模型(有的系统支持人机交互功能)。其娱乐性、舒适度及自然性也较好。

体显示通常是将三维物体分割为点阵或一系列二维图像，再依次扫描，利用人眼的视觉暂留效应形成立体图像。图1.15(a)中是把三维物体分割为点阵再依次扫描，图中所示立方体是添加了发光物质的透明荧光体，两束不可见波长的光聚焦到同一点进行激发，从而发出可见光，对立方体中每点依次扫描即可形成立体图像。图1.15(b)中是把三维物体分割为二维图像再依次扫描，以半圆形显示屏作为投影面，将它高速旋转，在空间形成一个半球形成像区域，在旋转的过程中将半圆形显示屏像素有规律的点亮，由于人眼视觉暂留效应从而观看到空间连续的三维图像。体显示可供多个观看者同时从不同角度观看到同一显示图像的不同侧面，且兼顾了人眼的调节和会聚特性，不会引起视觉疲劳。

小结

本节按基本工作原理是否为双目视差将三维立体显示分为两大类，并对各种三维立体显示器的结构、基本原理以及优缺点进行了简要分析。和一些研究方向不同，三维立体显示具有百家争鸣、百花齐放的特点，各种三维立体显示各具特点，各有各的使用场合；除了目前热门的全景视频、增强现实研究领域外，3D成像的概念也已应用到了立体电影、音像制品、手机、立体电视、户外媒体等产品与服务中。与其他新兴研究方向一样，立体显示技术还不完全成熟，还有很多未知的认识需要探讨，更高性能的器件和系统等待研制并产业化。我们也可以期待，在未来，人们可以脱去沉重的HMD轻松观看全景视频。

1.2 全景图像拼接技术

近年来，随着计算机技术的快速发展，图像融合技术发展越来越广泛和深入，对具有较大视域全景图像的需求也越来越迫切。全景图像拼接作为新兴技术，短短几年得到了快速发展，受到研究者越来越多的关注。目前全景图像已经成为计算机仿真、计算机视觉模拟、图

像处理和计算机特效以及虚拟现实研究中的热点和关键技术，在地质勘测、军事侦查、医学微创手术、航空航天以及视频会议等多个领域发挥着重要作用。

在之前章节中提到，目前全景图像/视频主要通过多相机拼接式全景视觉系统拍摄，由于相机多角度的特性，这类系统在产生原始图像后，需要通过全景图像/视频拼接技术进行后处理才能生成完整、统一的全景内容。

所谓图像拼接是指把多个单一图像融合成一幅图像。具体地说，全景拼接是将使用多个摄像机对同一个场景在不同角度拍摄得到的多个图像进行校正、去噪、匹配、融合，最终构建成一个质量高、清晰、边缘平滑、分辨率高的图像。图像拼接主要有以下几个步骤：相机标定、图像坐标变换、图像畸变校正、图像配准与融合。其中，图像配准与融合是图像拼接成功的关键。

全景拼接实现过程

(1) 相机标定

全景视觉系统中安装设计，以及摄像机之间的差异，会造成图像/视频之间有缩放（镜头焦距不一致造成）、倾斜（垂直旋转）、方位角（水平旋转）的差异，这类物理差异需要预先标定和校准，才能得到一致性好的图像，便于后续图像拼接。相机标定主要是求解相机的内参数矩阵，用于后续图像处理。该矩阵主要与相机的运动方式有关，常用求解方式是张正友标定法。该方法精度高，又避免了传统标定法对标定器材要求高、操作过程繁琐的约束。

相机的运动方式与成像结果之间的关系见下图：

名称	相机运动示意	图像变化结果	图像变换
平移			平移变换
变焦			放缩变换
水平旋转			投影变换
垂直旋转			投影变换
绕轴旋转			旋转变换

图 1.16 相机的运动方式与成像结果之间的关系

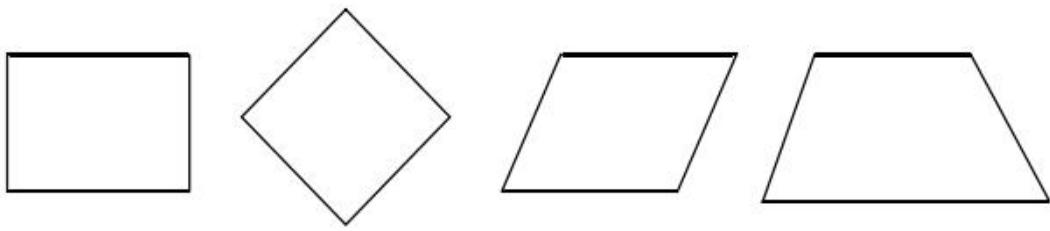
(2) 图像坐标变换

在实际应用中，全景图像的获得往往需要摄像机以不同的位置排列和不同的倾角拍摄。例如由于机载或车载特性，相机的排列方式不尽相同，不能保证相机在同一面上，如柱面投影不一定在同一个柱面上，平面投影不一定在同一平面上；另外为了避免出现盲区，相机拍

摄的时候往往会向下倾斜一定角度。这种情况比较常见，而且容易被忽略，直接投影再拼接效果较差。因而有必要在所有图像投影到某个柱面（或平面）之前，需要根据相机的位置信息和角度信息来获得坐标变换后的图像。

理论上只要满足静止三维图像或者平面场景的两个条件中的任何一个，两幅图像的对应关系就可以用投影变换矩阵表示，换句话说只要满足这其中任何一个条件，一个相机拍摄的图像可以通过坐标变换表示为另一个虚拟相机拍摄的图像。

为了确定图像序列的空间变换关系，需要确定图像对应关系的模型。平移变换模型、刚性变换模型、仿射变换模型和投影变换模型是目前常用的几何图像变换模型，图 1.17 即为几种几何变换模型示意图。



(a) 原始形状

(b) 刚性变换

(c) 仿射变换

(d) 投影变换

图 1.17 图像变换模型示意图

一般情况下，8 参数的透视投影变换最适合描述图像之间的坐标关系，其中 8 参数的矩阵为 $[m_0, m_1, m_2; m_3, m_4, m_5; m_6, m_7, 1]$ ，相应几何关系便用矩阵的形式描述出来：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = M \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.1)$$

各参数对应的相机运动表示如下：

表 1.1 透视投影变换参数与相机运动的关系

参数	对应摄像机的运动及成像结果
m_2	x 方向位移
m_5	y 方向位移
m_0, m_1, m_3, m_4	缩放，旋转，剪切
m_6, m_7	梯形失真 (x 方向和 y 方向形变)，线性调频

如图 1.18 显示的是相机向下倾斜一定角度拍摄图像的情况，这个角度与 m_6 和 m_7 具有对应关系，只需要对 8 参数矩阵求逆后进行坐标变换，即可获得校正图像，。

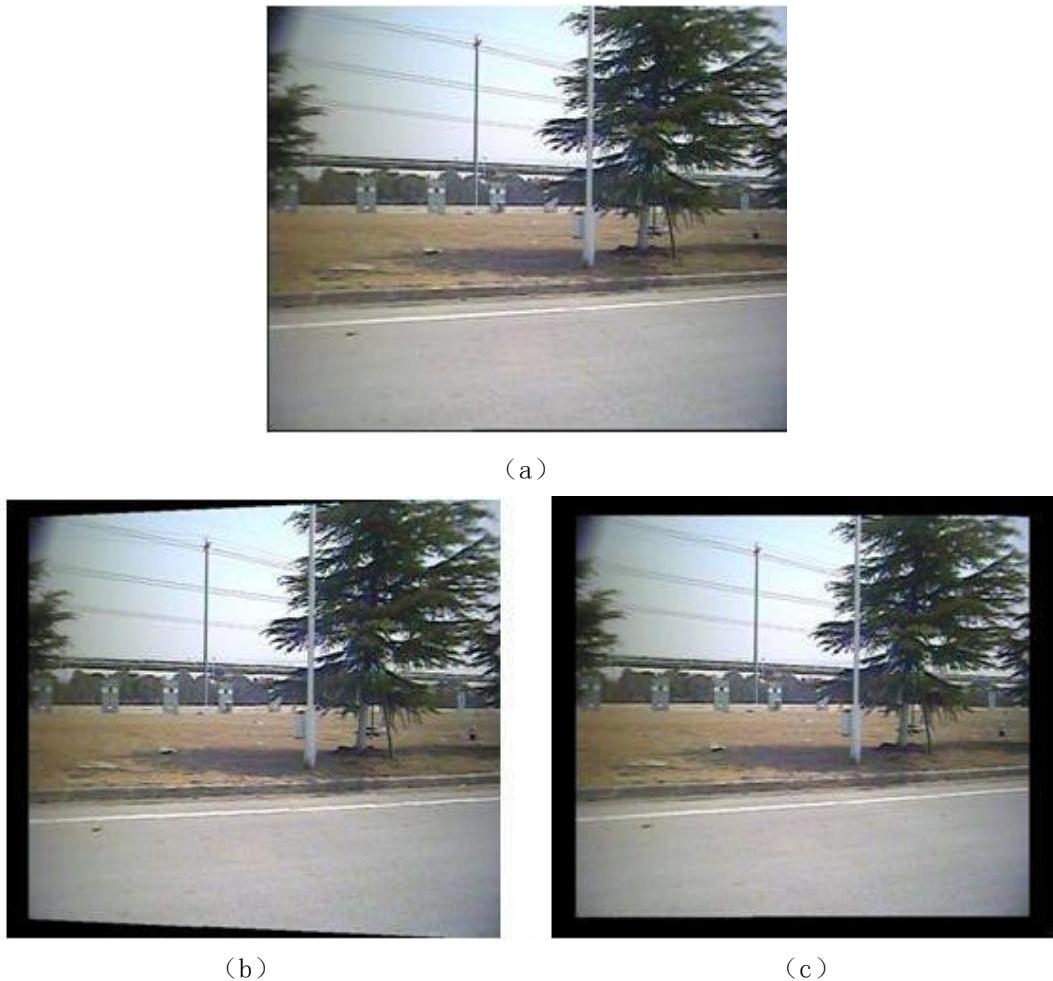


图 1.18 (a) 原始图像; (b) x 方向形变效果; (c) 倾斜校正后效果

(3) 图像畸变矫正

由于制造、安装、工艺等原因，镜头存在着各种畸变，为了提高摄像机拼接的精度，在进行图像拼接的时候必须考虑成像镜头的畸变。一般畸变分为内部畸变和外部畸变，内部畸变是由于摄像本身的构造为起因的畸变，外部畸变为投影方式的几何因素起因的畸变。镜头畸变属于内部畸变，由镜头产生的畸变一般可分为径向畸变和切向畸变两类。径向畸变就是集合光学中的畸变像差，主要是由于镜头的径向曲率不同而造成的，有桶形畸变和枕形畸变两种。切向畸变通常被认为是由于镜头透镜组的光学中心不共线引起的，包括有各种生成误差和装配误差等。一般认为，光学系统成像过程中，径向畸变是导致图像畸变的主要因素。径向畸变导致图像内直线成弯曲的像，且越靠近边缘这种效果越明显。根据径向畸变产生的机理，对视频图像进行校正。效果如图 1.19 (b) 所示，经过校正的图像，其有效像素区域缩小，一般可通过电子放大的方式进行校正，如图 1.19 (c) 所示。

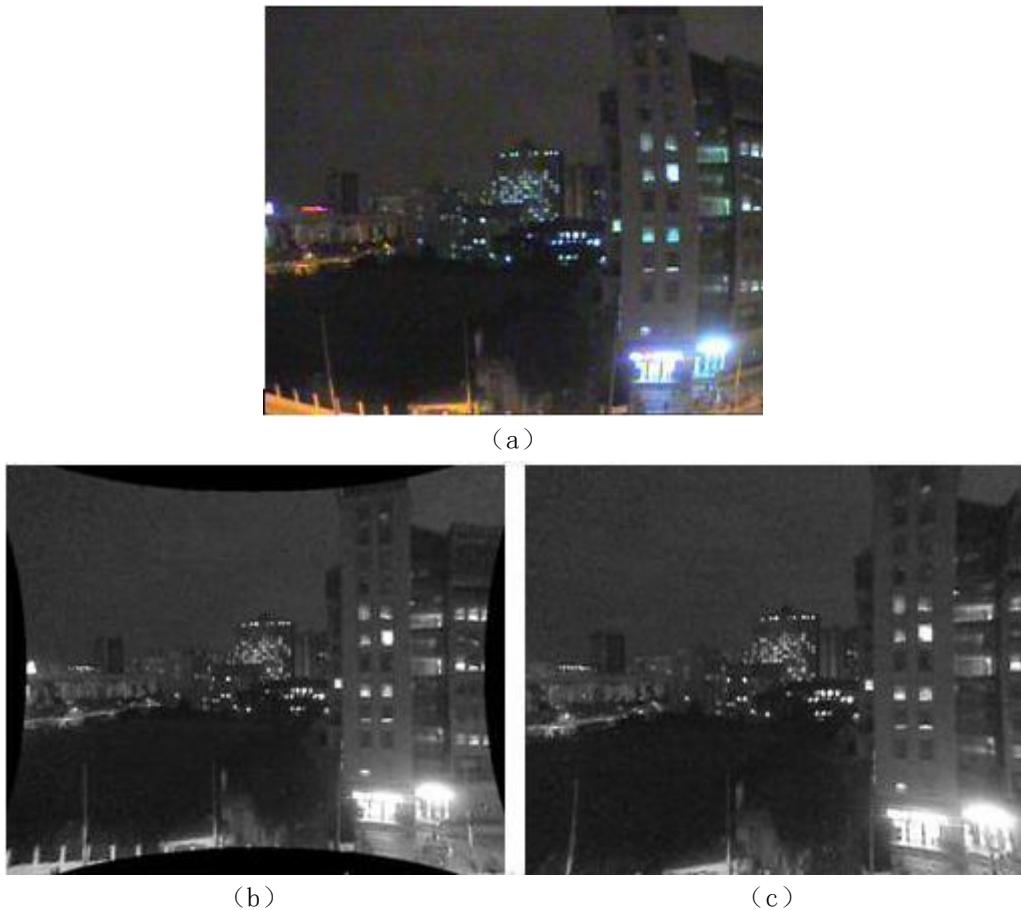


图 1.19 (a) 原始采集图像; (b) 经过径向失真校正的图像; (c) 经过放大的图像

(4) 图像配准与融合

全景拼接过程的最后一步，也是最为关键的一步是图像配准与融合。图像配准与融合在全景图生成、360° 全景相机以及 VR 全景领域有非常多的应用，常用的图像缝合工具有 Microsoft 的 ICE、PTGui、开源软件 Hugin 等，基于视频的拼接可以参考 VideoStitch、StitchHD 以及 stitching_with_cuda。

一般的图像配准与融合算法步骤可以描述为：

- 1) 定义映射模型，常用的包括：球面、柱面、平面，其中球面映射应用最为广泛；
- 2) 根据输入图像，提取特征点，对特征进行匹配，得到输入图像之间的映射关系 T ；
- 3) 根据映射关系 T 进行图像的 Warp 变换，对齐图像；
- 4) 包括利用颜色调整来消除图像间的色差，和采用图像融合来消除拼缝。

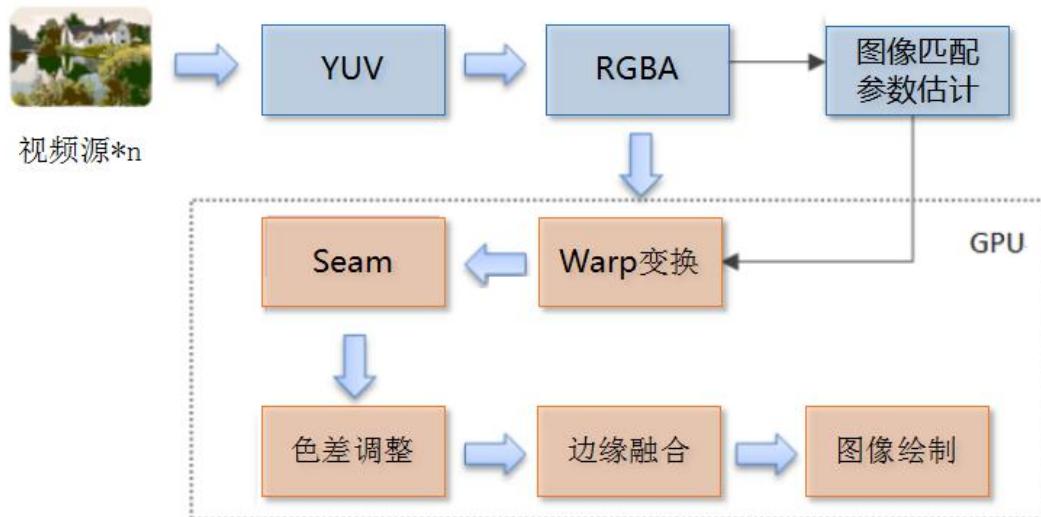


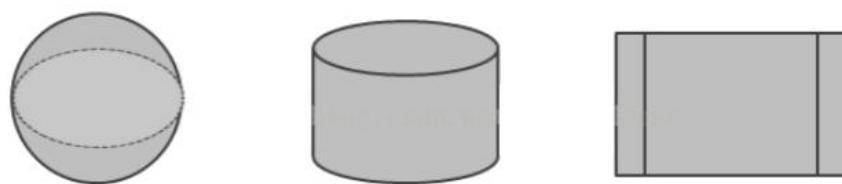
图 1.20 图像融合算法流程

图像投影变换

由于每幅图像是相机在不同角度下拍摄得到的，所以它们并不在同一投影平面上，如果对重叠的图像直接进行无缝拼接，会破坏实际景物的视觉一致性。所以需要先对图像进行投影变换，再进行拼接。一般有平面投影、柱面投影、立方体投影和球面投影等。

平面投影就是以序列图像中一幅图像的坐标系为基准，将其图像都投影变换到这个基准坐标系中，使相邻图像的重叠区对齐，称由此形成的拼接为平面投影拼接；柱面投影是指采集到的图像数据重投影到一个以相机焦距为半径的柱面，在柱面上进行全景区的投影拼接；球面投影是模拟人眼观察的特性，将图像信息通过透视变换投影到眼球部分，构造出一个观察的球面；立方体投影是为了解决球面映射中存在的数据不宜存储的缺点，而发展出来的一种投影拼接方式，它适用于计算机生成图像，对实景拍摄的图像则比较困难。

投影模型可以看作是用于图像映射的载体，相当于二维图像映射到三维空间的一种变换，如下图所示：



(a) 球面投影模型 (b) 柱面投影模型 (c) 平面投影模型

图 1.21 不同类型的投影模型

其对应拼接效果如下图所示：



图 1.22 不同投影模型的拼接效果

可以看到，不同投影模型对应的拼接效果是有所区别的，因而选择合适的投影模型非常重要。映射模型应与图像采集场景以及应用方式相匹配，一般对于水平拼接而言，采用柱面投影模型重现的效果最佳，而对于 360° 全景，球面映射或者立方体（多面体）映射的效果更好。由于上图中采用的是三张平行拍摄的图片，垂直方向张角较小，因此球面模型与柱面模型的拼接效果差异并不大。

特征点提取与匹配

由于特征点的方法较容易处理图像之间旋转、仿射、透视等变换关系，因而经常被使用，特征点包括图像的角点以及相对于其领域表现出某种奇异性的兴趣点。Harris 等人提出了一种角点检测算法，该算法是公认的比较好的角点检测算法，具有刚性变换不变性，并在一定程度上具有仿射变换不变性，但该算法不具有缩放变换不变性。针对这样的缺点，Lowe 提出了具有缩放不变性的 SIFT 特征点。

SIFT 特征检测主要包括以下 4 个基本步骤：

- 1) 尺度空间极值检测：搜索所有尺度上的图像位置。通过高斯微分函数来识别潜在的对于尺度和旋转不变的兴趣点；
- 2) 关键点定位：在每个候选的位置上，通过一个拟合精细的模型来确定位置和尺度。关键点的选择依据于它们的稳定程度；
- 3) 方向确定：基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向。所有后面的对图像数据的操作都相对于关键点的方向、尺度和位置进行变换，从而提供对于这

些变换的不变性;

4) 关键点描述: 在每个关键点周围的邻域内, 在选定的尺度上测量图像局部的梯度。这些梯度被转换成一种表示, 这种表示允许比较大的局部形状的变形和光照变化。



(a) 极值点初步检测 (b) 精确定位

图 1.23 SIFT 特征检测

特征匹配则用来计算图像之间的映射关系（采用 RANSAC 或者概率模型），得到每个匹配图像对之间的单应矩阵，结合上一步的映射模型，可以得到最终的图像变换序列 $T_1 T_2 T_3 \dots$ 。

不管使用何种图像检测算子，都需要在图像序列中找到有效的特征匹配点。图像的特征点寻找直接影响图像拼接的精度和效率。对于图像序列，如果特征点个数大于等于 4 个，则很容易自动标定图像匹配点；如果特征点很少，图像拼接往往不能取得较为理想的效果。

在计算映射变换之后，实际上相当于就得到了图像之间的全景变换关系，或者叫做相机变换参数（参照 OpenCV 里的 `detail::CameraParams`），对于视频拼接来讲，这个变换参数通常是不变的。

此外，也可以利用手动方式进行调整，只要确保图像之间对齐即可，因此特征的提取与匹配并不是必选项。

图像配准

图像拼接中的一个关键步骤是配准，即将两幅待拼接图像对齐，找到两幅图像相对的位置关系。配准的目的是基于匹配的特征点，根据几何运动模型将图像转换到同一个坐标系中。在多幅图像配准的过程中，采用的几何运动模型主要有：平移模型、相似性模型、仿射模型和透视模型。

图像的平移模型是指图像仅在二维空间发生了方向和方向的位移，如果摄像机仅仅发生了平移运动，则可以采用平移模型。图像的相似性模型是指摄像机本身除了平移运动外还可能发生旋转运动，同时，当场景存在缩放时，还可以利用缩放因子对缩放运动进行描述，因此，当图像可能发生平移、旋转、缩放运动时，可以采用相似性模型。图像的仿射模型是一个6参数的变换模型，即具有平行线变换成平行线，有限点映射到有限点的一般特性，具体表现可以是各个方向尺度变换系数一致的均匀尺度变换或变换系数不一致的非均匀尺度变换及剪切变换等，可以描述平移运动、旋转运动以及小范围的缩放和变形。图像的透视模型是具有8个参数的变换模型，功能是完成从二维坐标到二维坐标之间的线性变换，且保持二维图形的“平直性”和“平行性”。其可以完美地表述各种变换，包括平移，缩放，翻转，旋转和剪切，是一种最为精确变换模型。

采用矩阵的形式可以描述为：

$$p' = H * p \quad (1.2)$$

$$\text{其中 } H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$$

矩阵 H 被称为投影变换矩阵。对上式进行分解化简，即是通常所谓的单应性变换：

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1} \quad (1.3)$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1} \quad (1.4)$$

图像配准变换是一项耗时的操作，相当于对里面每一个像素点进行一次变换，像素之间的操作相互独立，因此操作通常放在 GPU 上来并行处理，对于 CUDA 来讲，透视变换相当于将输入图像的索引坐标值映射到纹理坐标。

亮度与颜色均衡处理

相机和光照强度的差异会造成一幅图像内部，以及图像之间亮度的不均匀，拼接后的图像会出现明暗交替的现象，极大地影响融合效果。因而在融合前，还需进行亮度与颜色均衡处理操作，通常的处理方式是通过相机的光照模型，校正一幅图像内部的光照不均匀性，然后通过相邻两幅图像重叠区域之间的关系，建立相邻两幅图像之间直方图映射表，通过映射表对两幅图像做整体的映射变换，最终达到整体的亮度和颜色的一致性。

具体的亮度与颜色均衡处理方法有很多，如自动白平衡方法，也可以手动调整色温，其基本思路都是通过统计每幅图的亮度或颜色区间分布，将不同的亮度、颜色调整到与参考颜色一致的空间内。

比较常用的是 Reinhard 方法，将图像 I 转换到 lab 空间（降低三原色之间的相关性），通过图像的统计分析，利用目标图像 I' 的均值及标准差进行线性调整，公式描述为：

$$\left\{ \begin{array}{l} L = (l - \bar{l}) * \frac{\sigma_l}{\sigma_{l'}} + \bar{l}' \\ A = (a - \bar{a}) * \frac{\sigma_a}{\sigma_{a'}} + \bar{a}' \\ B = (b - \bar{b}) * \frac{\sigma_b}{\sigma_{b'}} + \bar{b}' \end{array} \right. \quad (1.5)$$

线性变换（方差作为斜率）能够保证源图像能够与目标图像在 lab 颜色空间具有近似的均值和方差。通常我们可以选定一副图像作为调整基准，当然也可以计算需要变换的所有图像的均值作为一个目标值。

全景图像融合与生成

图像融合则是将配准后的图像合成为一张大的拼接图像。待融合图像已配准好且像素位宽一致，综合和提取两个或多个多源图像信息。两幅（多幅）已配准好且像素位宽一致的待融合源图像，如果配准不好且像素位宽不一致，其融合效果也不好。

高效的图像融合方法可以根据需要综合处理多源通道的信息，从而有效地提高了图像信息的利用率，系统对目标探测识别的可靠性及系统的自动化程度。其目的是将不同相机以及传感器所提供的信息加以综合，消除多相机信息之间可能存在的冗余和矛盾，以增强影像中信息透明度，改善解译的精度、可靠性以及使用率，以形成对目标的清晰、完整、准确的信息描述。这诸多方面的优点使得图像融合在医学、遥感、计算机视觉、气象预报及军事目标识别等方面的应用潜力得到充分认识，尤其在计算机视觉方面。

一般情况下，图像融合由低到高分为三个层次：数据级融合、特征级融合、决策级融合。数据级融合也称像素级融合，是指直接对传感器采集得到的数据进行处理而获得融合图像的过程，它是高层次图像融合的基础，也是目前图像融合研究的重点之一。这种融合的优点是保持尽可能多的现场原始数据，提供其它融合层次所不能提供的细微信息。

像素级融合中有空间域算法和变换域算法，空间域算法中又有多种融合规则与方法，如逻辑滤波法，灰度加权平均法，对比调制法等；变换域中又有金字塔分解融合法，小波变换法。其中的小波变换是当前最重要，最常用的方法。

其中，变换域中的拉普拉斯金字塔法可以理解为通过对相邻两层的高斯金字塔进行差分，将原图分解成不同尺度（频率）的子图，对每一个之图（对应不同频带）进行加权平均，得到每一层的融合结果，最后进行金字塔的反向重建，得到最终融合效果。

在特征级融合中，保证不同图像包含信息的特征，如红外光对于对象热量的表征，可见光对于对象亮度的表征等等。

决策级融合主要在于主观的要求，同样也有一些规则，如贝叶斯法，D-S 证据法和表决法等。

融合算法常结合图像的平均值、熵值、标准偏差、平均梯度；平均梯度反映了图像中的微小细节反差与纹理变化特征，同时也反映了图像的清晰度。

关于图像融合，目前已有很多现成的开源代码，这里不再过多描述。融合的关键在于选择用于融合的子图像区域部分，要注意图像过大导致的效率问题，也要避免图像较小带来的信息缺失现象。

OpenCV 的代码是图像/视频融合比较好的入门材料，可以作为参考，OpenCV 的流程框架图如下：

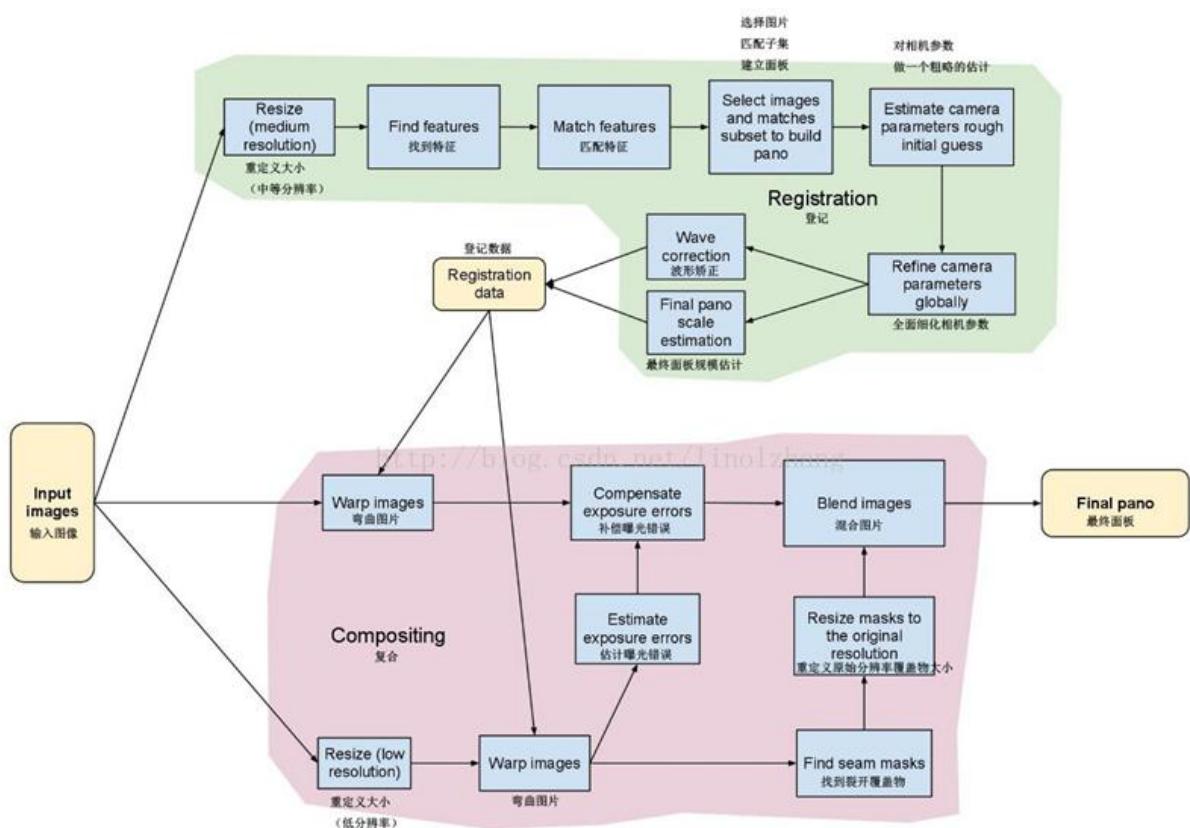


图 1.24 OpenCV 框架

1.3 开源软件

实际上，全景视频不仅可以用现实中的 3D 摄像机进行拍摄，也可以直接用三维软件进行渲染做出全景 CG（Computer Graphics）。对于最终可观看的全景视频的实现，可以通过将球型摄像机拍摄的或 360° 多角度拍摄的视频直接映射到一个球面上，也可以将拍摄的视频通过 EPR 映射将球形画面映射到矩形画面中，最后再将映射的画面贴回包裹着虚拟摄像机的球体模型表面。这样便能够完成全景视频的制作。下面将介绍几种能实现全景视频制作的软件。

Unity3D

Unity3D 是由 Unity Technologies 开发的一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等内容的多平台综合型游戏开发工具，是一个全面整合的专业游戏引擎。



图 1.25 Unity3D

MovieTexture 是 Unity 引擎内部自带，能够满足全景视频播放的组件，使用 MovieTexture 播放全景视频的效果较好，而且操作相对方便。但是，MovieTexture 只能播放 ogv 和 ovf，而且质量比较低。如果要转成高质量的视频，则文件尺寸比较大。

使用 MovieTexture 组件播放全景视频的方法是：

1. 在当前场景中新建一个 3D 的 Sphere 球体，将主相机的位置放在球体中心点处。
2. 将 ovf 或者 ogv 格式的视频文件放到 Resources 目录或者其子目录下，例如放在 Resources/Videos 目录下，通过以下代码来获取视频资源：

```
//不必带后缀名
MovieTexture tex = Resources.Load<MovieTexture>("Videos/video");
```

3. 获取 Sphere 对象上的 Renderer 组件，并将视频载入得到的 movTexture 纹理，传给 Renderer 的 material.mainTexture 属性，并设置视频纹理的播放模式：

```
//设置当前对象的主纹理为电影纹理
```

```
_parenTrans.GetComponent<Renderer>().material.mainTexture = movTexture;
```

```
//设置电影纹理播放模式为循环
```

```
movTexture.loop = true;
```

4. 通过以上步骤便完成了视频的加入以及播放前的所有准备，那么接下来要做的就是开始播放、暂停和停止视频等操作，较为简单，分别调用 MovieTexture 的接口即可：

```
//开始播放
```

```
movTexture.Play();
```

```
//暂停播放
movTexture.Pause();
//停止播放
movTexture.Stop();
```

运行时，查看结果可以发现视频被赋值到材质球中作为纹理，通过旋转相机 Y 轴的角度，可以 360 度地观看视频，这就是全景视频播放的一个过程。

CINEMA 4D

德国 MAXON 公司出品的 Cinema 4D，是一套整合 3D 模型、动画与算图的高级三维绘图软件，一直以高速图形计算速度著名，并有令人惊奇的渲染器和粒子系统，其渲染器在不影响速度的前提下，使图像品质有了很大提高，可以面向打印、出版、设计及创造产品视觉效果。Cinema 4D 使用的是 Python 语言。

在 Cinema 4D 中可以使用全景摄像机来渲染全景视频：

```
Camera = c4d.BaseObject(5103)
Camera[c4d.CAMERAOBJECT_SPC_ENABLE] = 1
```

上述代码表示摄像机的 ID 为 5103。全景摄像机并没有专门构建出一个对象出来，而是在原有的标准摄像机的基础上扩展了全景功能。所以在调用了标准摄像机对象以后，第二行代码激活了全景摄像机选项。

全景摄像机有很多个参数，比如 FoV 辅助参数，映射参数等。我们主要对映射模式参数进行设置。全景摄像机的映射模式主要有两种：Lat-Long 映射和立方体映射，Lat-Long 映射也就是经纬度映射（EPR），它将球面上的各个点映射到了一个二维矩形面上，这是渲染全景图时经常采用的一个方法，这种方法能够比较方便地和传统视频处理及编码技术进行结合。但是这种方法会造成视野顶部和底部较为明显的失真，接缝明显。而立方体映射（六面体映射）则很好地解决了这个问题，将画面渲染到围绕观察点的六个垂直的面上，形成立方体包裹，保证了顶部和底部较小的失真。

VRay

VRay 是由 chaosgroup 和 asgvis 公司出品的一款高质量渲染软件。基于 V-Ray 内核开发的有 VRay for 3ds max、Maya、Sketchup、Rhino 等诸多版本，为不同领域的优秀 3D 建模软件提供了高质量的图片和动画渲染，方便使用者渲染各种图片。VRay 渲染器提供了一种特殊的材质——VrayMt1。在场景中使用该材质能够获得更加准确的物理照明（光能分布），更快的渲染，反射和折射参数调节更方便。VrayMt1 中可以应用不同的纹理贴图，控制其反射和折射，增加凹凸贴图和置换贴图，强制直接全局照明计算，选择用于材质的双向反射分布函数（BRDF）。

与其他的渲染器相比，VRay 的灵活性、易用性更好，有着焦散之王的美誉。VRay 还包括了其他增强性能的特性，包括真实的 3d Motion Blur（三维运动模糊）、Micro Triangle Displacement（级细三角面置换）、Caustic（焦散）、通过 VRay 材质调节完成 Sub-surface scattering（次表面散射）的 sss 效果、和 Network Distributed Rendering（网络分布式渲染）等等。VRay 特点是渲染速度快，目前很多制作公司使用它来制作建筑动画和效果图，就是看中了它速度快的优点。VRay 渲染器有 Basic Package 和 Advanced Package 两种包装形式。Basic Package 具有适当的功能和较低的价格，适合学生和业余艺术家使用。Advanced Package 包含有几种特殊功能，适用于专业人员使用。目前市场上有很多针对 3DSMAX 的第

三方渲染器插件，Vray 就是其中比较出色的一款。主要用于渲染一些特殊的效果，如次表面散射、光迹追踪、焦散、全局照明等，也可以使用 Vray 插件在 3DMAX 中渲染全景图片。Vray 是一种结合了光线跟踪和光能传递的渲染器，其真实的光线计算创建专业的照明效果，可用于建筑设计、教学等多个领域。

1.4 典型全景相机介绍

随着虚拟现实行业的兴起，原本只常见于机器人制造、工业、监控等领域的全景摄像机被越来越多地被应用到生活娱乐领域，大量新的厂商进入全景相机领域，市面上的全景拍摄器材日益丰富。目前，国外的 Nokia、Google、GoPro、Facebook、Jaunt、三星、松下、柯达、尼康等知名企业以及国内强氧、得图、Insta360、UCVR、兹曼等企业都在布局 VR 全景摄像机。

就目前的使用情况而言，全景摄像机按照镜头类别主要可以分为鱼眼镜头全景摄像机、多镜头全景摄像机两类。

(1) 鱼眼全景摄像机

鱼眼全景摄像机由单传感器配套特殊的超广角鱼眼镜头组成，工作原理在之前章节已经提到过，依赖图像校正技术还原图像。由于性价比高，此前这类产品在市场中占据主流份额，常被用于工业、监控等领域，比如考场、大厦的安保监控等。然而由于鱼眼镜头的特殊性，画面边缘畸变部分难以达到高清晰度，对于生活娱乐的市场而言，并不能满足需求。因此随着虚拟现实消费市场拓展，其所占市场份额有所下降。



图 1.26 鱼眼摄像头

(2) 多镜头全景摄像机

多镜头全景摄像机是通过多个传感器配合特制的镜头组合实现全景功能的。由于多镜头拼接全景摄像机的各个传感器捕获的都是常规矩形图像，因此无需进行矫正操作。而其缺点在于大部分多镜头全景相机需要配套可实现画面无缝拼接的算法或软件，且在组装时对镜头

视场角与安装位置的设定都有严格要求。另外，由于多镜头全景摄像机相当于多个镜头+传感器的组合，因此成本自然会有所增加，性价比上低于鱼眼镜头摄像机，对于普通消费者而言略显昂贵。不过，多镜头全景摄像机的适用性相当广泛，可以支持各类生活、娱乐，甚至影视级别的全景视频内容拍摄，当前网络上的各类VR直播、VR短片基本都是使用这类全景摄像机拍摄而成的。

以下将介绍几款典型的多镜头全景摄像机：

GoPro Omni

GoPro 是一款小型可携带固定式防水防震运动相机，被广泛应用在冲浪、滑雪、极限自行车及跳伞等极限运动上，被认为是“极限运动专用相机”。2016 年 4 月，GoPro 正式发布全景相机 GoPro Omni VR，这款相机采用了立方体结构，内部集成了 6 颗该公司的 Hero4 Black 摄像头。据 Gopro 方面称，该设备采用了特殊的技术，使得 6 颗摄像头之间能够实现“像素级别的同步”。



图 1.27 Gopro Omni

技术规格：

各摄像机分辨率/帧速率（球面分辨率）

- 2.7K 4:3 / 30 fps (7940x3970)
- 2.7K 4:3 / 25 fps (7940x3970)
- 1440p / 60 fps (5638x2819)
- 1440p / 50 fps (5638x2819)

视频格式：

H.264 codec、.mp4 文件格式，45Mb/s (Protune™ 60Mb/s)

音频格式：

48kHz，原始 PCM

媒体：

6 张 microSD 卡

连接/控制：

- 6 根迷你 USB 连接线，用于摄像机充电和卸载
- 1 个 Smart Remote，用于远程控制

软件：

Omni 特有阵列摄像机固件

电源：

- 6 颗可充电锂离子电池（规格：1160mAh, 3.8V, 4.4Wh）
- 1 个 12V 电源输入端（规格：2.5mm x 5.5mm, 7A）

物理尺寸：

120mm x 120mm x 120mm

Surround 360

Surround360 是一个用于摄制和渲染 3D (立体) 360 视频和照片的硬件和软件系统。其硬件部分的全景摄像机形状类似飞碟，共采用 17 部超高清摄像机环绕构成，并配有基于网络的软件，可在 360 度范围内捕捉图像和自动呈现。这款设备能够连续工作而不会出现过热现象，拍出的视频分辨率最高可达 8K。其全景视频可以在 Gear VR、Oculus 等虚拟现实头盔以及 Facebook 的应用上观看。这款相机的易用性在业界备受赞誉，其全部配件均可在市面上购买到。目前这款设备的售价为 3 万美元，为了降低售价，Facebook 方面已将 Surround 360 的硬件组装图和后期拼接软件放到了 Github 上，向全球开发者开放其成果，希望更多的开发者共同打造出更优质、更强大的全景摄像工具。



图 1.28 Surround 360

Jaunt VR ONE

Jaunt ONE 是专为摄制高质量立体 360° 电影虚拟现实体验而设计的专业级摄像系统。作为行业内首个电影级虚拟现实摄录一体机，Jaunt ONE 是 VR 创作先行者心目中的理想产品。经历了两年以上的高强度研究和开发，Jaunt ONE 不但拥有酷炫的外形，更拥有专为 VR 定制的光学系统，支持高质量全景音视频捕捉。

其主要特点为：

- 1) 24 目同步全局快门
- 2) 最高支持 120 帧/秒 (FPS)
- 3) 3D 全景覆盖
- 4) 单眼分辨率最高达 8K × 4K
- 5) 动态曝光调节
- 6) 单个模块自动/手动曝光调节
- 7) 远程相机操控及实时监测



图 1.29 Jaunt ONE

除了上述专用于拍摄全景视频的一体机，常见的攒机方案有：

多目 GoPro 方案

多目 Gopro 黑狗 4 方案是目前性价比最高，也是最受拍摄团队喜爱的方案。一方面，它可以根据需求任意组装，6 目、7 目、8 目、10 目、12 目甚至 14 目均可实现，没有任何限制。支架既可以从各平台上购买成品，也可以自己 3D 打印按需定制。另一方面，每台黑狗的成本在 3000 元左右，因此入门级设备 2~3 万即可拥有，且操作简单，成本低廉，上手门槛低。国内的天狗全景、莱瑞特、Upane 等团队都使用多目 Gopro 解决方案，它可以拍摄绝大部分非专业类全景视频和部分专业类全景视频。

强氧方案

强氧是目前国内较专业的全方位影像服务电商。其 2 代解决方案由 10 台 Drift Ghost-S 拼接而成，上两台，下两台，中间六台。Drift Foream Ghost-S 在长期运行、直播、散热、耗电量、稳定性等方面较 Go Pro 有一定的优势。其 3 代方案 Argus Panoptes Pro 在 2 代产品上更进一步，全景相机采用了环绕 8 颗，上方 1 颗摄像头的 9 目设计。感光元件采用的 M43 系统，支持拍摄 4K 60 帧的全景视频。强氧解决方案被广泛应用于各类大型活动直播，曝光率较高，官网商城的全景直播解决方案套餐价格为 12 万。

红龙拼接

如果需要拍摄影视级别的全景内容，则需要使用更加专业的解决方案。国内已有不少 VR 影视制作团队开始使用红龙拼接的方案。使用 4 台 RedDragon 面向四个方向进行拍摄，可以达到 24K 100FPS 的录制标准，也可以进行 30FPS 标准的 4K 直播。不过这套方案造价也

相当昂贵，价格近 200 万元，普通的消费和商业全景内容无需使用这类设备，否则处理巨大的素材源也将成为一大难题。其他基于红龙的 VR 电影解决方案也大致如此。6 目，7 目，8 目等等。

HeadcaseVR

2016 年，HeadcaseVR 团队正式公布了全新移动 VR 拍摄方案。这个创业团队来自好莱坞。专门从事 VR 电影拍摄工作。该方案主要采用 17 目 Codex Action Cameras。Codex Cam 有 12bit RAW 的记录体系和 13.5 档的高动态。采用 2/3 英寸的 CCD 传感器，单分辨率 1920×1080 ， $23.98\text{fps} - 60\text{fps}$ 的帧率表现。头部尺寸为 $45\text{mm} \times 42\text{mm} \times 53\text{mm}$ ，因而外观十分小巧，同时也配备专业的采集设备来实现录制。这个团队还定制了更适合移动 VR 视频拍摄的移动工具，虽然看上去像一台布满电池和采集器以及供电元件的轮椅，结构复杂又臃肿。不过这台移动设备解决了在 VR 视频拍摄中由移动产生的位移偏差及抖动问题。目前拍摄的负责人是 Marc Dando。

HypeVR

Hype 采用将 14 个 RedDragon 拼合的方式一步到位地实现了真正达到电影级 VR 设备的方案，该方案运用了激光雷达传感技术，由 Velodyne 公司设计，RedDragon 是 6K 级的。HypeVR 最终拼接完成可以达到 16K 90FPS 标准。

NextVR

跟 HypeVR 类似，NextVR 采用的也是 RED DRAGON 6K。但是他们选择了 6 台的拼接方案，三个方位，每个方位安放两台。虽然机器只有六台，但是价格依旧不菲。可以初步估算下：RedDragon 3 万美元一台，佳能 8-15mm f/4L 鱼眼镜头 1400 美元一个，监视器是 Red Pro，1600 美元一个，……。基于该方案，4K 直播可以轻松实现。

J2VR-极图全景

中国的团队 J2VR 目前也给出了经过几代更新后的电影级 VR 解决方案。采用 4 台 RedDragon 分别对四个方向进行拍摄采集。最终达到 24K 录制 100FPS 的标准。官方声称打造这个方案花费了 180 万重金，这让大众拭目以待。

Google JUMP

它实际上仅是一个支架。需要配合 16 台 GoPro 和专业软件才能实现全景视频采集和拼接。由于并没有设计向上或者向下的镜头，该方案只能在垂直范围内采集 120° 的图像。但是它的优势在于高性价比，花费约几万块就可以实现 360° 的 VR 视频拍摄。

本章参考资料：

- [1] 王敏, 周树道, 张水平, 黄峰. 全景立体成像技术浅述 [J]. 信息技术, 2014(05):24-27+30.
- [2] 肖潇, 杨国光. 全景成像技术的现状和进展 [J]. 光学仪器, 2007(04):84-89.
- [3] 王琼华, 王爱红. 三维立体显示综述 [J]. 计算机应用, 2010, 30(03):579-581+588.
- [4] 韩伟. 3D 影视图像技术泛论 [J]. 有线电视技术, 2009, 16(11):79-81.
- [5] 丁剑飞, 刘永进. 三维立体显示技术综述 [J]. 系统仿真学报, 2008, 20(S1):132-135.
- [6] 郑华东, 于瀛洁, 程维明. 三维立体显示技术研究新进展 [J]. 光学技术, 2008(03):426-430+434.

- [7] 冯传岗. 摘掉眼镜看 3D 浅谈影视图像的 3D 显示技术 [J]. 数码影像时代, 2012(11):44-51.
- [8]https://www.chinatmic.com/360_Panoramic/show/92.html
- [9]<http://www.elecfans.com/instrument/581578.html>
- [10]<https://www.xzbu.com/8/view-4775061.htm>
- [11]<https://wenku.baidu.com/view/3424ef6fa9956bec0975f46527d3240c8547a15a.html>
- [12] Ishiguro H, Yamamoto M, Tsuji S. Omni-Directional Stereo[M]. IEEE Computer Society, 1992.
- [13] Shum H Y, Szeliski R. Stereo Reconstruction from Multiperspective Panoramas: IEEE, US6639596[P]. 2003.
- [14] 丁艳. 全方位视觉技术的综述[J]. 科技信息:科学教研, 2007(34):236-237.
- [15] 王健, 张振海, 李科杰, 等. 全景视觉系统发展与应用[J]. 计算机测量与控制, 2014, 22(6).
- [16]<http://baijiahao.baidu.com/s?id=1582008687252798978&wfr=spider&for=pc>
- [17]http://www.sohu.com/a/108274423_423652
- [18]<https://blog.csdn.net/linolzhang/article/details/54377060>
- [19]<https://wenku.baidu.com/view/9c9f3acded630b1c58eeb551.html>
- [20] 赵书睿. 全景图像拼接关键技术研究[D]. 电子科技大学, 2013.
- [21] 江铁, 朱桂斌, 孙奥. 全景图像拼接技术研究现状综述[J]. 重庆工商大学学报(自然科学版), 2012, 29(12):60-65+71.
- [22] 宋宝森. 全景图像拼接方法研究与实现[D]. 哈尔滨工程大学, 2012.
- [23]<https://baike.baidu.com/item/SIFT/1396275?fr=aladdin>
- [24]<https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E8%9E%8D%E5%90%88/625475?fr=aladdin>
- [25]<http://www.goprochina.cn/cameras1/omni---%E4%BB%85%E5%8C%85%E5%90%AB%E5%A5%97%E7%9B%92/MHDHX-007.html>
- [26]https://www.sohu.com/a/109076657_411731
- [27]<https://github.com/facebook/Surround360>
- [28]<https://www.jauntvr.cn/technology/>

第二章 全景视频呈现技术

2.1 HMD 的基本组成及典型设备

HMD 简介



图 2.1 各式各样的 HMD 设备

头戴式显示器 (Head Mount Display)，缩写为 HMD，指佩戴于头部的显示装置。HMD 有许多用途，包括游戏，航空，工程、医学等。

典型的 HMD 具有一个或两个小型显示器，并通过将透镜和半透镜嵌入眼镜（也称为数据眼镜）实现。显示单元是小型化的，可能包括阴极射线管 (CRT)、液晶显示器 (LCD)、硅上液晶 (LCos)、有机发光二极管 (OLED) 等。一些供应商会采用多个微显示器来达到提高总分辨率和扩大视野的目的。

大多数 HMD 仅显示计算机生成的图像 (CGI)，也称为虚拟图像，或来自真实物理世界的实时图像。还有一部分 HMD 允许 CGI 叠加在真实世界的视图上，称为增强现实或混合现实 (AR/MR)。真实世界视图与 CGI 的结合可以通过将 CGI 投射到部分反射镜并直接观察现实世界的方法来完成，这种方法通常称为光学透视；也可以通过接收来自摄像机的视频并以电子方式与 CGI 混合的形式完成。这种方法通常称为视频透视。

基本性能参数

a) 立体图像显示能力：为了让用户形成立体图像的错觉，双目 HMD 向每只眼睛输入不同的图像。在物体与眼睛距离较大时，物体在每只眼睛内形成的图像趋近相同；而对于距离较小的范围，两只眼睛的视角明显不同，所以通过 CGI 系统产生两个不同的视觉通道是有必要的。

b) 瞳距 (IPD)：指两只眼睛之间的距离，瞳距的设定对于头戴式显示器而言非常重要。

c) 视场 (FoV)：人类的 FoV 约为 220 度，但大多数的 HMD 远远低于此值。通常，更大的视场会带来更强的沉浸感和更好的情境感知。消费级别的 HMD 通常提供约 30–40° 的 FoV，而专业 HMD 提供的 FoV 在 60° 至 150° 之间。

d) 分辨率 (Resolution)：在 HMD 中通常表示为像素总数或每度数的像素数 (PPD)。60 PPD 通常被称为人眼上限分辨率。超过该分辨率，视力正常的人不会注意到增加的额外分辨率。目前的 HMD 通常只能提供 10–20 PPD，但微显示器的发展可以逐渐增加这一数值。

e) 双目重叠，双眼共有的区域：双目重叠是视觉深度和立体感的基础，它允许人们感知物体的远近。人类的双目重叠约为 100° 。而 HMD 提供的双目重叠程度越大，立体感越强。一般会以度数指定重叠范围或以百分比表明每只眼睛相对于另一只眼共同的虚拟 FoV 占比。

HMD 关键技术

想要营造真正令人沉浸的 VR 体验，需要攻克一系列技术难关：宽视角、高分辨率、像素低存留、高刷新率、全局显示、光学透镜和校准、低延迟，以及运动追踪。

视角越宽，沉浸感、临场感会越强，此外，尽管人们很难分辨出视线边缘的文字和图形，但这些视觉线索对移动、平衡、环境感知而言仍然至关重要。

随着视野变宽，像素将被拉伸，视野中每一度对应的像素将会减少，图像则变得粗糙。以 $1K \times 1K$ 分辨率的显示屏为例，当它在 HMD 中，以 110 度视角展现给使用者时，像素密度只有普通电视机的七分之一，人眼上限分辨率的十分之一，这甚至还不如在 320×200 分辨率下的原始 PC 游戏清晰。若想得到较好的临场感，至少需要以 $1080p$ 分辨率显示图像。

对于 HMD，由于双眼距离显示屏越近，它们之间的相对位移越快，因而如果像素存留时间较长，前庭动眼反射（VOR）将导致眼球移动时的图像变得模糊。如果不加以解决，在头部移动时，几乎无法清晰阅读文字，对 $1K \times 1K$ 分辨率的 HMD 来说，像素存留时间不能超过 3 毫秒，而且像素密度越高，存留时间应越短，相对应的，刷新率就需要提高。在任何场景下都保持高刷新率不容易。同时随着分辨率提高，需要有更强大的硬件和与之紧密配合的软件来支持高刷新率。

人眼对视觉上的异常非常敏感。有些 HMD 在静止时观看的图像尚可接受，但一摆头，临场感却荡然无存。HMD 需要控制焦距、畸变、像差等多种因素，这也是照相机为了获得高质量图片需要拥有复杂光学镜头的原因。而受空间和重量限制，HMD 中，每只眼睛前最多只能有一到两个镜片。对此，Oculus 重新发明了 HMD 的制造方式，利用计算机技术预先处理图像畸变，而不再倚赖造价高昂的光学镜片。

延迟是 HMD 技术的核心，高延迟是导致眩晕，破坏沉浸感的罪魁祸首。图像变化与用户实际移动之间的延迟接近 20 毫秒，就能产生相当不错的体验。对于 HMD，延迟存在于 6 个层面：a) 运动捕获；b) 数据和命令通过 USB 线缆传入 PC 或手机；c) 游戏引擎响应命令后调动 GPU；d) GUP 生成一帧新图像；e) HMD 开始显示像素；f) 图像完全显示。这一点在之后章节中仍会提到。

运动追踪技术同样是使用户相信自己真实地处于虚拟世界的关键技术。它可以通过追踪头部的运动状态实时更新渲染的场景，这与我们在真实世界中观看周围环境非常类似。高速的惯性测量单元（IMU）被用于快速的头动追踪。它结合了陀螺仪、加速度计（或磁力计，类似手机中使用的重力感应装置），可以精确测量转动的变化。头部运动追踪非常重要，因为人类的感知系统对运动非常敏感，如果在头部移动时图形显示出现延迟，那么就会破坏沉浸感，甚至引起恶心不适。因而，虚拟现实的 IMU 设备必须快速追踪头部运动，同时相应软件的性能也应匹配。只有当立体渲染和运动追踪结合得更好时，才能使得图形刷新帧率足够高，虚拟现实体验才能达到真正意义上的沉浸感。

就目前的情况来看，VR 行业的发展态势良好，包括 HypeVR、NextVR 在内的许多团队始终致力于 VR 各环节的技术研究。而像 Facebook、Samsung、HTC 等知名厂商也已着眼于 VR，并于近年带来了许多先进、便捷的产品。接下来便将对主流的 HMD 产品进行简要介绍。

主流 HMD —— 高端性能的主机+头盔

a) Vive-Pro



图 2.2 Vive-Pro 设备

- 单眼分辨率 1440 x 1600，双眼分辨率为 3K (2880 x 1600)
- 刷新率 90Hz
- 视场角 110 度
- 关键传感器：SteamVR 追踪、G-sensor 校正、gyroscope 陀螺仪、proximity 距离感测器、瞳距感测器

尽管相比初代产品，性能上提升了很多，但原先的HTC Vive并没有被抛弃，HTC为Vive和Vive Pro推出了无线适配器。正如人们所期望的那样，这个小插件可以夹在HMD后部，无需将接线连回PC端。此外，Vive Pro后部的新增配件和定位盘极大地提升了长期沉浸式体验的舒适感。

b) Pimax 8K VR



图 2.3 Pimax 8K

不出所料，从奥运广播测试平台引入8K等现象来看，8K也进入了VR世界。

虚拟现实领域对于8K的研究一直很少也很难进行，直至中国创业公司Pimax为其开辟了道路。该公司在2017年筹集了超过400万美元的资金，承诺向VR提供模块化方案，允许赞助商从一系列选项中选择硬件规格，包括无线，手柄和眼部追踪等相关的硬件。

实际上，Pimax并没有提供人们熟悉的电视屏幕（7,680 x 4,320分辨率）显示的8K，而是相对较低的7,680 x 2,160的分辨率。然而，这仍然是业界领先的分辨率方案（单目4K，90Hz刷新率）。但分辨率提高的同时，该公司也面临着8K VR内容、兼容的PC硬件紧缺的问题。

Pimax已经表明，对于那些追求更高分辨率VR的用户，Pimax 还研究了更强的 Pimax 8K X设备，这个版本经过特别设计，需要 NVIDIA GTX 1080 Ti来驱动，然而仅为使用该显卡，Pimax就需要为每台设备支付约1000英镑的成本。

除了8K分辨率的亮点，Pimax特隆风格的造型在同类产品中焕然一新。为了提升VR的沉浸效果，Pimax 8K采用了稍小于人类自然220度视场的200度FoV，是目前VR产品中最好的，同时减少了VR内容边缘的黑色边框。Pimax也承诺在正式出货之前实现15ms MTP的延迟。

c) Oculus Rift (DK2)



图2.4 Oculus Rift (DK2) 设备

- 低延迟视觉跟踪系统
- 刷新率超过 75Hz
- 视场角 100 度
- 单眼分辨率 960×1080

d) HMD Odyseey



图2.5 HMD Odyseey设备

- 分辨率 2880×1600
- 视场角 110 度
- 前面板有两个摄像头，用于实现 inside-out 定位追踪以及现实空间数据的收集
- 配备的手柄在运动时配合头显上的摄像头进行数据采集，从而进行定位。其优点在于成本不高而且定位效果不错；缺点在于容易受环境因素影响。

主流 HMD —— 手机+头盔

a) Samsung New Gear VR



图2.6 Samsung Gear VR设备

- 视场角101度
- 关键传感器：加速度传感器，陀螺传感器，接近传感器

b) Google DayDream



图2.7 Google DayDream设备

- 视场角90度
- 配备无线控制器，内置陀螺仪，可检测方向、行动以及加速，实现位置追踪

主流 HMD —— 一体机

a) 联想 Mirage Solo



图 2.8 联想 Mirage Solo

VR 技术成为科技主流的一个明显标志就是更多的制造商将产品线扩大至虚拟现实领域。联想便是其中之一，其在 CES 上发布了与谷歌合作的 VR 一体机 Mirage Solo。

“无需 PC 或智能手机，没有杂乱的接线”，Mirage Solo 基于 Daydream 平台融合了运动跟踪技术，采用联想 Mirage 相机，可在 180 度 FoV 范围内拍摄 VR 视频。此外，Mirage Solo 搭载骁龙 835 芯片，配备分辨率为 2560 x 1440 LCD 液晶显示屏，续航时间达 7 小时。

更为重要的是，Mirage Solo 采用了 Google WorldSense 六自由度追踪定位系统，配备三自由度的控制器。WorldSense 不仅能追踪用户头部的旋转角度，而且还可以追踪整个身体在佩戴 Mirage Solo 时的移动情况。

与 Google 合作的另一创新点是 VR180，VR180 是以用户为中心的 180 度视频格式，观看时可以通过转头来切换视角，视频边界则以黑边显示，相当于 360 视频切掉了一半。这一理念是由大部分用户只观察 VR 视频前面部分的习惯而来，目的是创造出更具身临其境的宽幅图像和视频，而无需使用拍摄 360 度视频的繁重设备。Mirage 相机是第一批 VR180 相机之一，契合 Google 捕捉 180 度全景图像和视频的新方式。双镜头可拍摄 4K 视频，用户可以在任何 VR 设备上观看三维视频，以及 YouTube 和 Google 照片。

b) Pico Neo 6Dof



图2.9 Pico Neo 6Dof设备

- 双眼分辨率：2880x1600

- 视场角：101°
- 无需视力调节，自适应瞳距
- 双目摄像头，用于头部6DoF空间定位
- 超声波定位传感器，用于手部6DoF空间定位
- 关键传感器：高精度九轴传感器

c) Vive-Focus



图2.10 Vive-Focus设备

- World-Scale 六自由度大空间追踪技术，高精度九轴传感器，距离传感器
- 刷新率 75Hz
- 视场角 110 度
- 搭配高精度九轴传感器操作手柄，更具交互性

浅析 Oculus 的成功

具有深度信息且持续的 3D 渲染是虚拟现实最重要的部分。为了达到这个效果，VR 设备都需要借助一个立体显示设备（即 HMD）。

过去，由于缺乏价格低廉并且长时间佩戴舒适的 HMD 设备，以致 VR 头显设备无法大范围进行推广。而 Oculus VR 团队改变了这个局面。他们在 2012 年推出了 Oculus Rift，这款产品具有立体显示功能并且内置头动追踪设备。这款设备不但轻便，而且售价只有几百美元。尽管 Oculus Rift (DK1) 的分辨率还很低，但这也足以引起一场产业风暴。新版本的 Oculus Rift (DK2)，分辨率、移动追踪性能和显示效果都有所提升。

Oculus Rift 都做了哪些革新性的工作呢？第一，为了产生深度信息，它为每个眼睛生成一张图片，这两张图片在视觉上有一些偏移量，这样就可以模拟人眼的视差。第二，为了产生更好的视觉效果，它通过桶形畸变技术，将图片扭曲从而模拟人眼的球形表面。

有了以上这些技术，Tuscany VR Demo 场景在虚拟现实设备中显示如图 2.11。



图 2.11 Oculus Rift 连接电脑以后通过电脑屏幕看到头显中的真实图像

2.2 渲染相关技术

2.2.1 渲染的定义

渲染过程试图利用有限数目的像素将图像空间连续函数表现成颜色，在电脑绘图中是指用软件从模型生成图像的过程。模型是用严格定义的语言或者数据结构对于三维物体的描述，它包括几何、视点、纹理以及照明信息。

假设我们在一个空的三维空间中要确定一个面，那么至少得有三个点，而一个物体就是由一个个的面构成，在图形学中一般用三角面代替。这是在我们假象的三维空间中，但是如果要在二维的显示器中看到该物体，便需要将该物体的三角面从三维空间渲染到二维空间中，如图 2.12 所示。

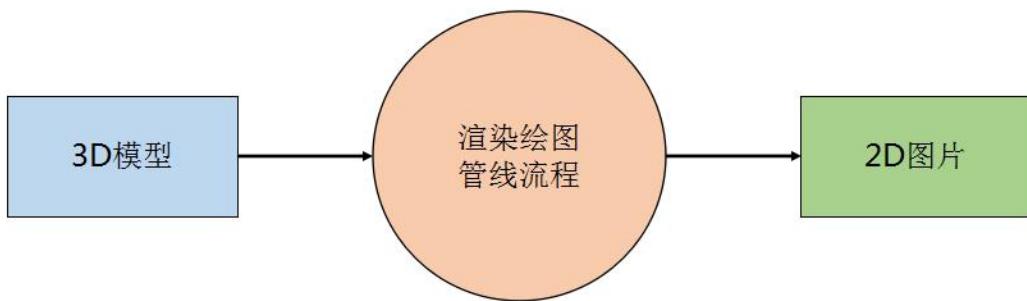


图 2.12 渲染图示：从 3D 到 2D

几何模型

我们首先需要一个虚拟世界来包含几何模型。为此，一个具有笛卡尔坐标的 3D 欧几里德空间就足够了。接着，令 R^3 表示虚拟世界，其中每个点都表示为一个三元组实值坐标： (x, y, z) 。虚拟世界的坐标系如图 2.13 所示。

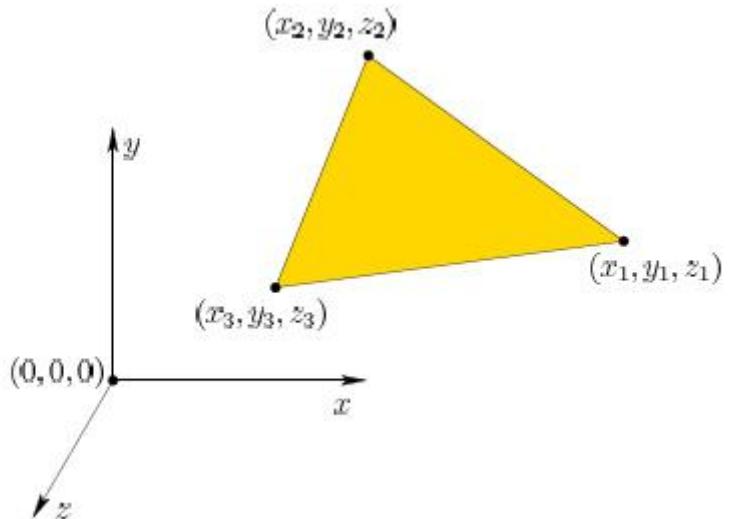


图 2.13 虚拟世界的几何模型

几何模型由 \mathbb{R}^3 中的曲面或实心区域组成，并包含无限多点。由于计算机中的表示是有限的，模型是根据基元来定义的，其中每个基元表示一组无限点。最简单和最有用的基元是一个 3D 三角形，如图 2.13 中的三角形。对应于“内部”所有点和三角形边界上的平面由三角形顶点的坐标完全指定。为了模拟虚拟世界中的复杂物体，我们可以将大量的三角形排列成一个网格，如图 2.14 所示。

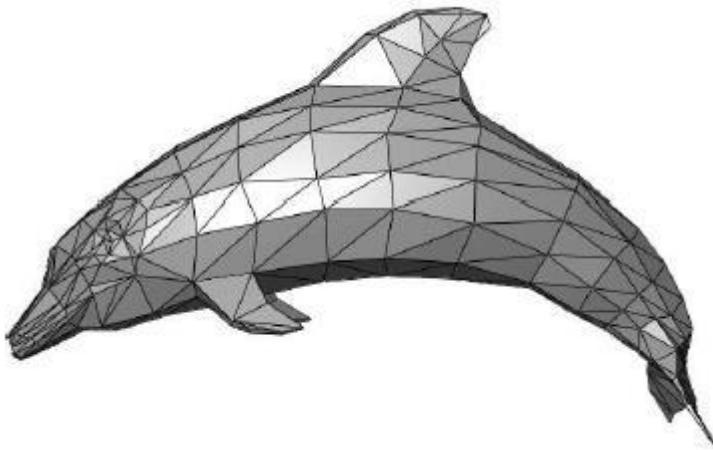


图 2.14：一只海豚的几何模型，由 3D 三角形的网格组成。（来自维基百科用户 Chrschn）

为什么是三角形

使用三角形是因为它们对于算法来说是最方便处理的，特别在硬件实施方面。GPU 的实现倾向于较小的表示，以便可以并行地将多个指令的紧凑列表应用于多个模型部分。当然，也可以使用更复杂的原型，如四边形，样条线和半代数曲面。这可能导致更小的模型尺寸，但通常会因处理这样的原型而带来更大的计算开销。例如，相比于两个 3D 三角形来说两个样条曲面很难确定是否碰撞。

物体顺序 vs 图像顺序

假设一个虚拟世界已经以三角形基元的形式被定义。在此基础上，便可以放置一双虚拟

的眼睛，并从一些特定的位置和方向观看虚拟世界。每个三角形都被正确地放置在虚拟屏幕上。接下来的步骤是确定哪些屏幕像素被变换后的三角形覆盖，然后根据虚拟世界的物理原理照亮它们。在该过程中必须检查一个重要的条件：对于每个像素，三角形是否对眼睛可见，还是会被另一个三角形的一部分阻挡？这种经典的可视性计算问题极大地使渲染过程复杂化。更一般的问题是确定虚拟世界中的任何一对点，连接它们的线段是否与任何物体（三角形）相交。如果发生交叉，则两点之间的视线可视性被阻止。渲染方法的主要区别就是如何处理可视性。

对于渲染，我们需要考虑物体和像素的所有组合。这表明了一个嵌套循环。解决可视性的一种方法是迭代所有三角形的列表并尝试将每个三角形渲染到屏幕上。这被称为物体顺序渲染，对于落入屏幕视场中的每个三角形，仅当三角形的相应部分比目前为止渲染的任何三角形更接近眼睛时才更新像素。在这种情况下，外部循环遍历三角形，而内部循环遍历像素。另一种方法称为图像顺序渲染，它颠倒循环的顺序：遍历图像像素，并且对于每一个像素，确定哪个三角形会影响其 RGB 值。为了实现这一点，进入每个像素的光波路径将通过虚拟环境被追踪。

2.2.2 计算机图形学中的渲染方法

为了从 3D 场景转换到 2D，场景中的所有物体都需要转换到几个空间。每个空间都有自己的坐标系。这些转换是通过一个空间的顶点转换到另一个空间的顶点来实现的。

光照 (lighting)，是这个阶段的另一个主要部分，是使用物体表面的法向量来计算的。通过摄像机的位置和光源的位置，可以计算出给定顶点的光照属性。

对于坐标系变换，我们从物体坐标系开始，每个物体都有自己的坐标系，这有利于几何变换，如平移，旋转和缩放。我们进入到世界坐标系，场景中的所有物体都具有统一的坐标系。下一步是转换到视图空间，即摄像机坐标系。想象一下：先在世界空间中放一个虚拟摄像机，然后进行坐标变换，使得摄像机位于视图空间的原点，镜头对准 z 轴的方向。现在我们定义一个所谓的视体 (view frustum)，它用来决定我们通过虚拟的 3D 摄像机所能看到的场景，只需要把这些内容渲染出来即可，如图 2.15 所示。

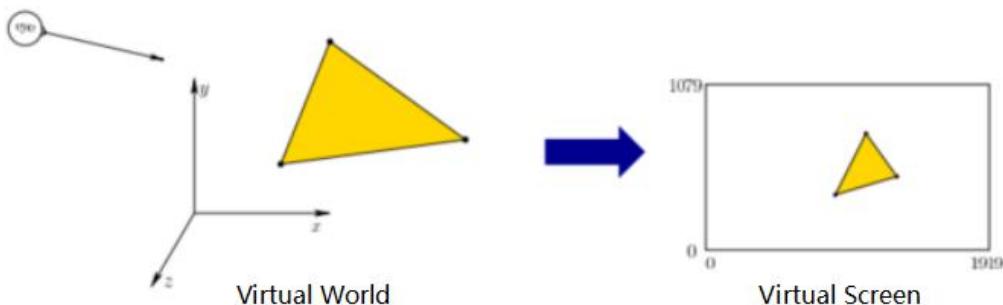


图 2.15 视角坐标变换

光栅化渲染方法

在现实世界中，三角形会将光线向各个方向散射。但是对于一台普通电脑，根本不可能去计算所有不同方向的光线，所以，计算机渲染图像，仅仅计算了那些散射到我们人眼方向的光线，CG 里面也叫摄像机方向，图形学中称之为视角方向。在渲染中，我们需要精确地知道一个光线是照射到一个指定的几何体上的，在该过程中也需要知道射入点的一些几何属性，比如面法线 (surface normal) 或者它的材质 (material)。大部分光线追踪器都包含测量

一个光线和多个物体的相交性，返回距离最近的对象之类的功能。一个光线追踪器应该对场景中的光照进行建模，除了描述光源的位置，也包括描述这些光的能量是如何分布在场景中的。因而假设在 3D 空间中添加一个摄像头，并在前端透视线点上放置一个屏幕网格，其中每个框是渲染图像的一个像素。我们只需画出与虚拟相机透视线点相交的一些光线。如果这些光线相交于我们的屏幕，那么屏幕就能知道我们观察该三角形的边界位置，如图 2.16 所示。

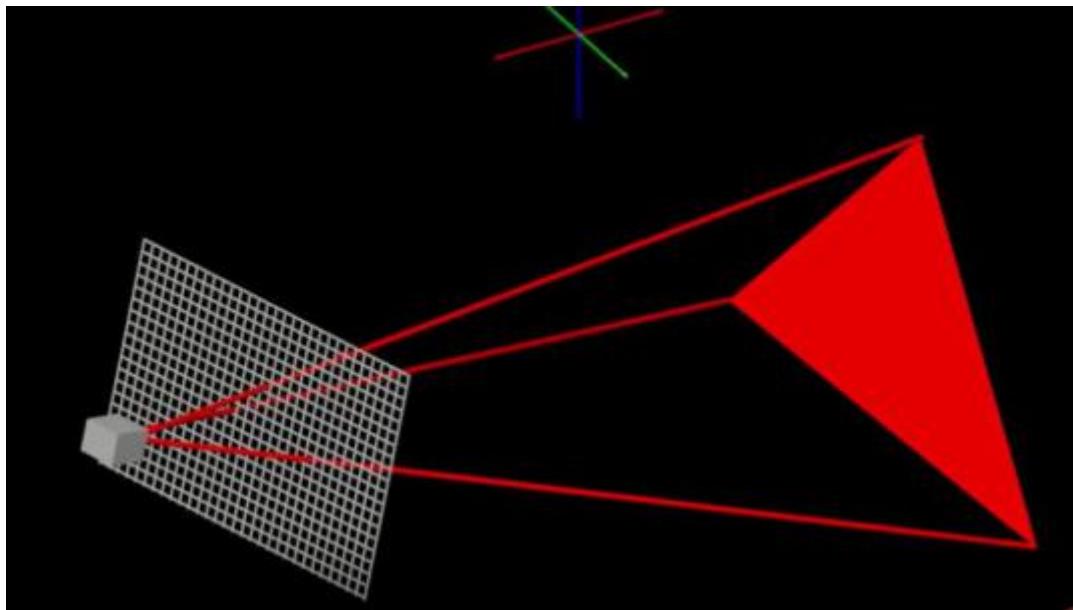


图 2.16 视角方向建模

知道边界以后，对边界与像素之间重叠的部分渲染像素，其余地方不做处理，便可以得到这个三角形的图像，如图 2.17 所示：

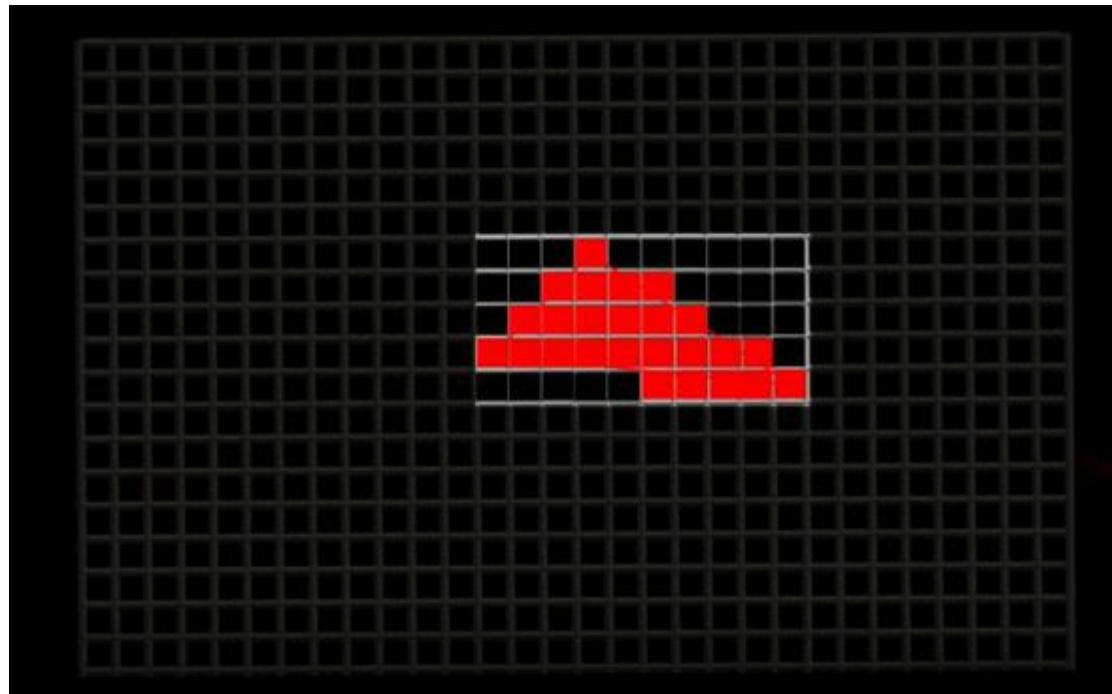


图 2.17 渲染成功的三角形

光线投射

光栅化是以物体为中心的，从相机中捕捉物体所发射到相机的光线，而 Ray casting 以

图像为中心，只考虑那些实际有用的光线，从虚拟相机开始，并通过相机向每一个像素都发射一条光线：

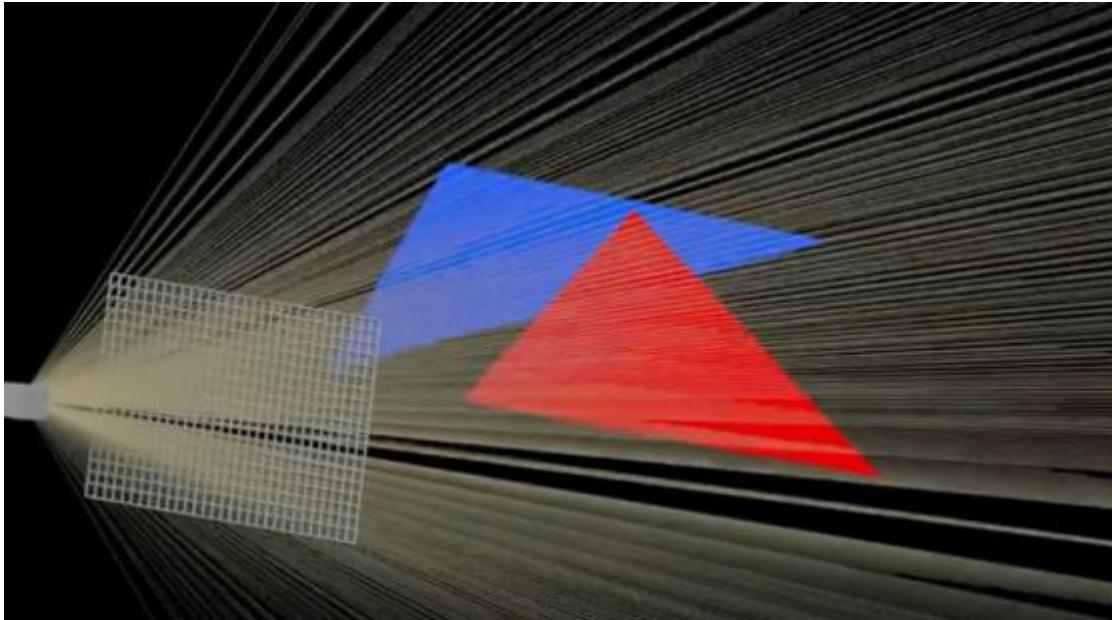


图 2.18 光线投射示意图

现在将判断每条射线是否射中了每一个三角形，如果一个光线遇到多个物体，取最近的那个点。如果忽略计算性能，计算观察光线在离开图像像素后碰到的第一个三角形非常简单直接的。从三角形坐标，焦点和射线方向（矢量）开始，封闭形式的解决方案包含来自解析几何的基本操作，包括点积，交叉积和平面方程。对于每个三角形，必须确定射线是否与其相交。如果不是，则考虑下一个三角形。如果相交，那么仅当交叉点比迄今为止遇到的最近交叉点更近时，交叉点才被记录为候选解决方案。在考虑所有三角形之后，最近的交点就被找到。虽然这种计算很简单，但是它也有很大的弊端，就是计算量过于庞大，因为该方法需要判断相当多的射线与物体的三角面是否相交。假设有一个 1000×1000 像素的图像，那么该方法就要计算 1,000,000 条光线是否和场景中一个多边形相交，对计算机来说，这计算相当地费时费力，即使目前在算法上有了很大的改进，仍然需要大量的计算。该方法比较常见的例子包括 BSP 树和 Bounding Volume Hierarchies。对几何信息进行排序以获得更高效的算法通常被归类为计算几何。除了从快速测试中消除许多三角形外，许多计算光线三角交点的方法也被开发以减少操作次数。其中最受欢迎的是 Möller-Trumbore 相交算法。

光线追踪

光线追踪的流行来源于它比其它渲染方法如扫描线渲染或者光线投射更加能够现实地模拟光线，像反射和阴影这样的一些对于其它算法来说都很难实现的效果，却是光线追踪算法的一种自然结果。光线追踪易于实现并且视觉效果很好。

当主光线接触一个表面时，通过在反射方向上绘制二次光线来绘制阴影光线，只要这个光线可以反射到光源，那么我们就知道是光源照亮了这个物体，如图 2.19 所示。如果我们发现光与表面之间存在物体，这时表面就处于阴影中，如图 2.20 所示，如此不断的反射被称之为递归射线追踪。

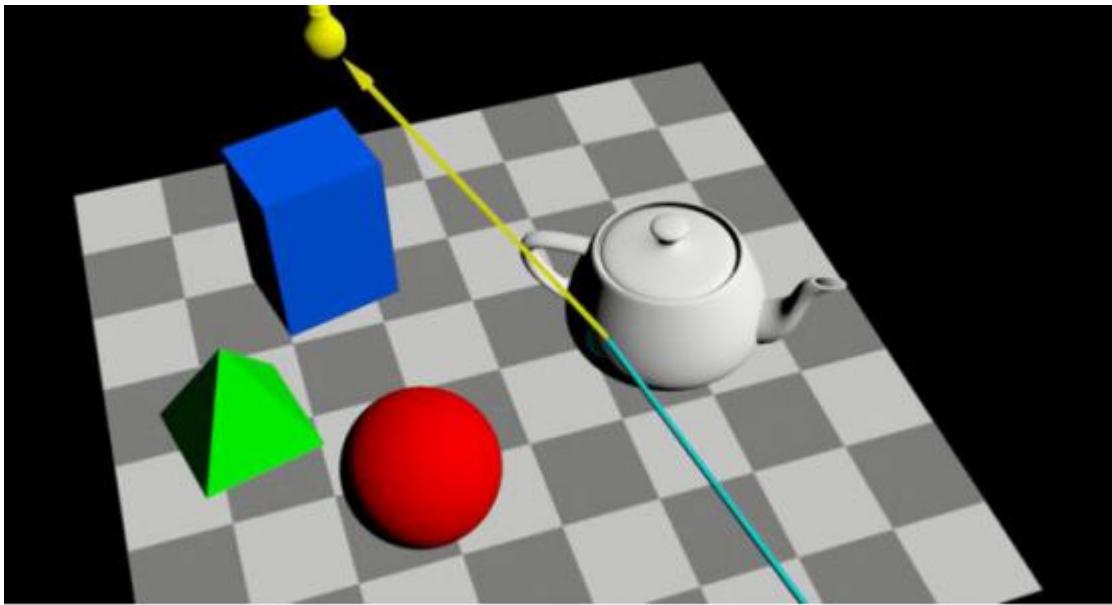


图 2.19 光线追踪绘制反射光线

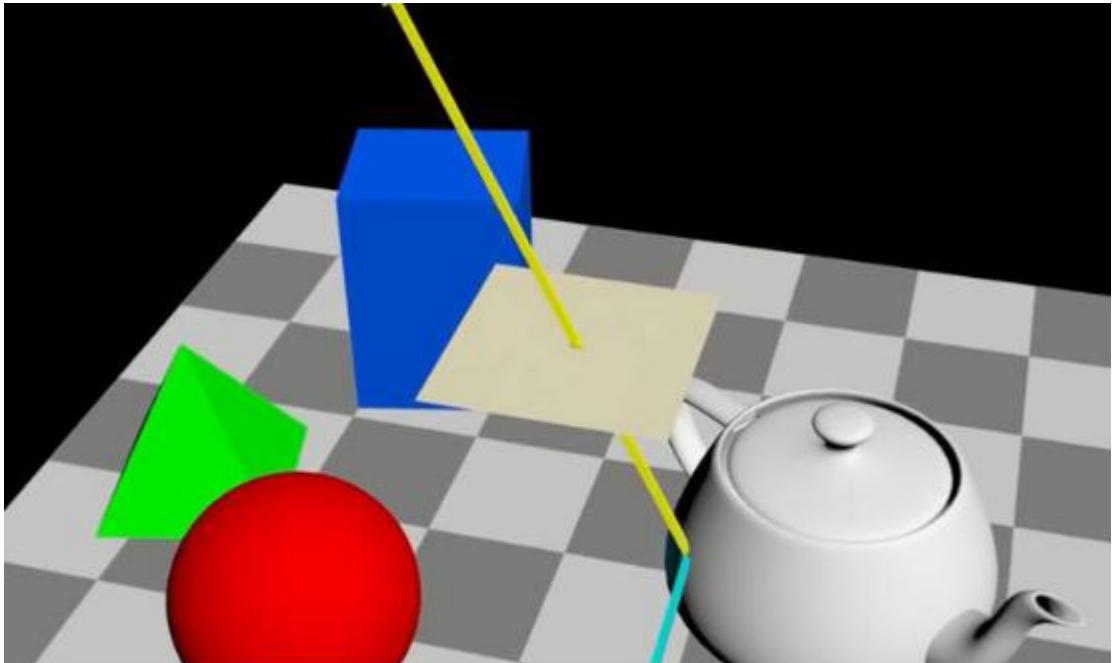


图 2.20 光与表面之间存在物体，至于阴影处

为了计算像素的 RGB 值，根据放置在虚拟世界屏幕上的焦点通过像素的中心绘制观察光线，该过程分为两个阶段：

1. 光线投射，定义了观察射线并计算虚拟世界中所有三角形之间的最近交点。
2. 阴影，根据照明条件和交点处的材料属性计算像素 RGB 值。

第一步完全基于虚拟世界几何。第二步使用虚拟世界的模拟物理。物体的材料属性和光照条件都是人工的，并且可以被选择来产生所需的效果。光线追踪的一个最大的缺点就是性能，扫描线算法以及其它算法利用了数据的一致性从而在像素之间共享计算，但是光线追踪通常是将每条光线当作独立的光线，每次都要重新计算。但是，这种独立的做法也有一些其它的优点，例如可以使用更多的光线以抗混叠现象，并且在需要的时候可以提高图像质量。尽管它正确地处理了相互反射的现象以及折射等光学效果，但是传统的光线追踪并不一定是最真实效果图像，只有在非常近似或者完全实现渲染方程的时候才能实现真正的真实效果图。

像。

2.2.3 光栅化

经过变换的顶点流按照顺序被送到下一个被称为图元装配和光栅化的阶段。首先，在图元装配阶段根据伴随顶点序列的几何图元分类信息把顶点装配成几何图元。这将产生一序列的三角形、线段和点。这些图元需要经过裁剪到可视平截体（三维空间中一个可见的区域）和任何有效的应用程序指定的裁剪平面。光栅器还可以根据多边形的朝前或朝后来丢弃一些多边形。这个过程被称为挑选（culling）。

经过裁剪和挑选剩下的多边形必须被光栅化。光栅化是一个决定哪些像素被几何图元覆盖的过程。多边形、线段和点根据为每种图元指定的规则分别被光栅化。光栅化的结果是像素位置的集合和片段的集合。当光栅化后，一个图元拥有的顶点数目和产生的片段之间没有任何关系。例如，一个由三个顶点组成的三角形占据整个屏幕，因此需要生成上百万的片段。片段和像素之间的区别变得非常重要。术语像素（Pixel）是图像元素的简称。一个像素代表帧缓存中某个指定位置的内容，例如颜色，深度和其它与这个位置相关联的值。一个片段（Fragment）是更新一个特定像素潜在需要的一个状态。之所以术语片段是因为光栅化会把每个几何图元（例如三角形）所覆盖的像素分解成像素大小的片段。一个片段有一个与之相关联的像素位置、深度值和经过插值的参数，例如颜色，第二（反射）颜色和一个或多个纹理坐标集。这些各种各样的经过插值的参数是来自变换过的顶点，这些顶点组成了某个用来生成片段的几何图元。你可以把片段看成是潜在的像素。如果一个片段通过了各种各样的光栅化测试，这个片段将被用于更新帧缓存中的像素。

光栅操作阶段根据许多测试来检查每个片段，这些测试包括剪切、alpha、模板和深度等测试。这些测试涉及了片段最后的颜色或深度，像素的位置和一些像素值（像素的深度值和模板值）。如果任何一项测试失败了，片段就会在这个阶段被丢弃，而更新像素的颜色值（虽然一个模板写入的操作也许会发生）。通过了深度测试就可以用片段的深度值代替像素深度值了。在这些测试之后，一个混合操作将把片段的最后颜色和对应像素的颜色结合在一起。最后，一个帧缓存写操作用混合的颜色代替像素的颜色。

2.2.4 贴图

当一个图元被光栅化为一个或多个片段时，插值、贴图和着色阶段就在片段属性需要的时候插值，执行一系列的贴图和数学操作，然后为每个片段确定一个最终的颜色。除了确定片段的最终颜色，这个阶段还确定一个新的深度，或者甚至丢弃这个片段以避免更新帧缓存对应的像素。

纹理贴图

在计算机图形学中，纹理贴图是使用图像、函数或其他数据源来改变物体表面外观的技术。例如，可以将一幅砖墙的彩色图像应用到一个多边形上，而不用对砖墙的几何形状进行精确表示。当观察这个多边形的时候，这张彩色图像就出现在多边形所在位置上。只要观察者不接近这面墙，就不会注意到其中几何细节的不足（比如其实砖块和砂浆的图像是显示在光滑的表面上的事实）。通过这种方式将图像和物体表面结合起来，可以在建模、存储空间和速度方面节省很多资源。

通过纹理贴图，一些重复的图案，如瓦片或条纹等可以在物体表面进行传播，如图 2.21 所示。更普遍的说，任何数字图像都可以映射到三角形上。重心坐标指的是图像中可以影响像素的点。图像，或是说“纹理”可以视为被绘制在三角形上；此外，为了更加真实，使物体有遮蔽效果，可以额外添加光照和反射属性。

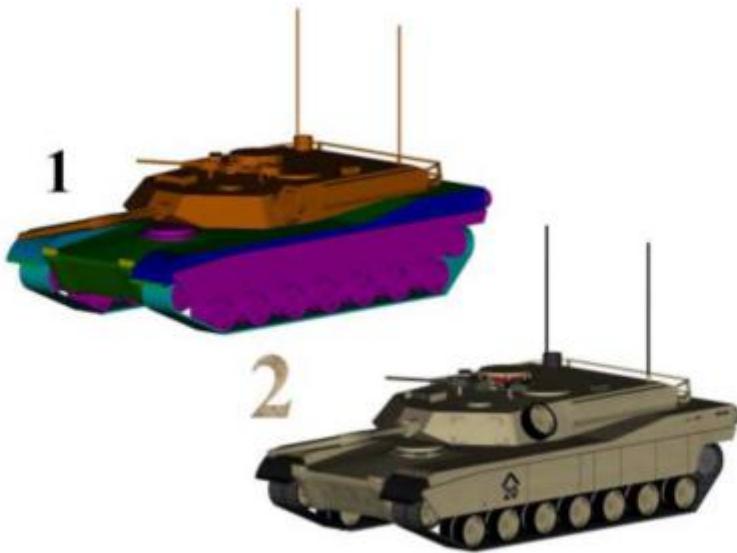


图 2.21 纹理贴图：将简单的图案或是整幅图像映射在三角形纹理中，然后在图像上进行渲染，相比于直接使用模型中的三角形纹理，这样可以提供更多细节。（图源于 Wikipedia）

纹理管线

简单来说，纹理（Texturing）是一种针对物体表面属性进行“建模”的高效技术。图像纹理中的像素通常被称为纹素（Texels），区别于屏幕上的像素。根据 Kershaw 的术语，通过将投影方程（projector function）运用于空间中的点，可以得到一组称为参数空间值（parameter-space values）的关于纹理的数值。这个过程就称为贴图（Mapping，也称映射），也就是纹理贴图（Texture Mapping，也称纹理映射）这个词的由来。纹理贴图可以用一个通用的纹理管线来进行描述。纹理贴图过程的初始点是空间中的一个位置。这个位置可以基于世界空间，但是更常见的是基于模型空间。因为若此位置是基于模型空间的，当模型移动时，其纹理才会随之移动。如图 2.22 为一个纹理管线（The Texturing Pipeline），也就是单个纹理应用纹理贴图的详细过程，而此管线有点复杂的原因是每一步均为用户提供了有效的控制。

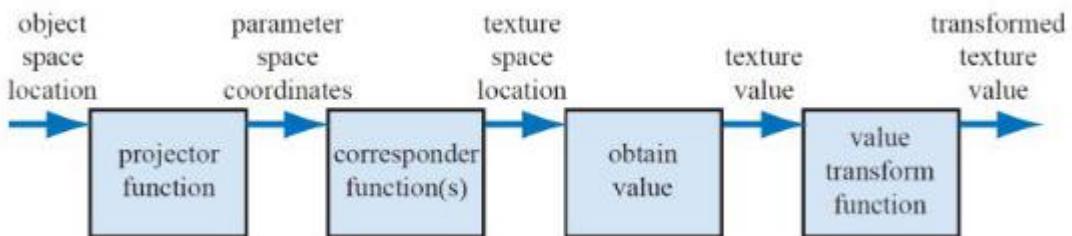


图 2.22 纹理管线流程图

第一步。通过将投影方程（projector function）运用于空间中的点，从而得到一组称为参数空间值（parameter-space values）的关于纹理的数值。

第二步。在使用这些新值访问纹理之前，可以使用一个或者多个映射函数（corresponder function）将参数空间值（parameter-space values）转换到纹理空间。

第三步。使用这些纹理空间值（texture-space locations）从纹理中获取相应的值（obtain value）。例如，可以使用图像纹理的数组索引来检索像素值。

第四步。再使用值变换函数 (value transform function) 对检索结果进行值变换，最后使用得到的新值来改变表面属性，如材质或者着色法线等等。

投影函数 The Projector Function

作为纹理管线的第一步，投影函数的功能就是将空间中的三维点转化为纹理坐标，也就是获取表面的位置并将其投影到参数空间。在常规情况下，投影函数通常在美术建模阶段使用，并将投影结果存储于顶点数据中。也就是说，在软件开发过程中，我们一般不会去用投影函数去计算得到投影结果，而是直接使用在美术建模过程中，已经存储在模型顶点数据中的投影结果。通常在建模中使用的投影函数有球形、圆柱、以及平面投影，也可以选其他一些输入作为投影函数。

映射函数 The Correspondent Function

映射函数 (The Correspondent Function) 的作用是将参数空间坐标 (parameter-space coordinates) 转换为纹理空间位置 (texture space locations)。我们知道图像会出现在物体表面的(u, v)位置上，且 u, v 值的正常范围在[0, 1]范围内。超出这个值域的纹理，其显示方式便可以由映射函数 (The Correspondent Function) 来决定。

法线贴图

另一种可行方法是法线贴图，是通过在三角形上人工改变表面法线来改变遮蔽的过程，尽管它不能在几何上实现。允许其变化的话，可以在物体上添加一个仿真的曲率，法线贴图的一个重要实例叫做凹凸贴图，通过不规则地扰动法线使得平坦的表面变得粗糙。如果法线具有纹理的话，那么在计算阴影之后，整个表面看上去会显得很粗糙。

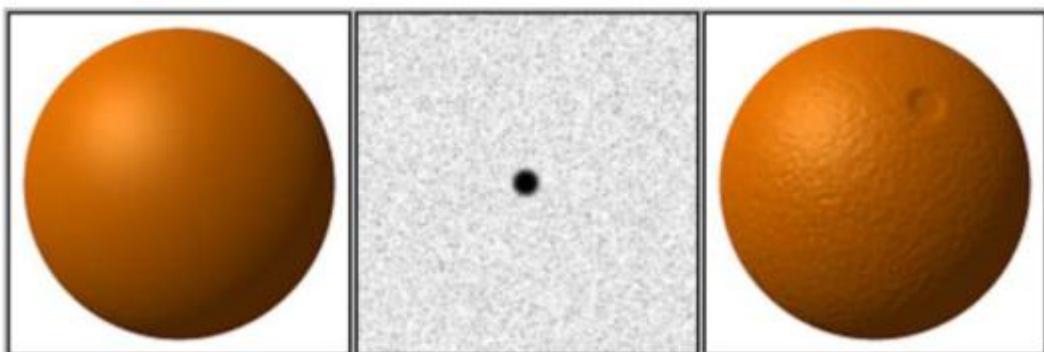


图 2.23 凹凸贴图：通过人工改变表面法线，阴影生成算法会产生一个看上去粗糙的表面（图源于 Brian Vibber）

凹凸贴图与其改进

凹凸贴图是指计算机图形学中在三维环境中通过纹理方法来产生表面凹凸不平的视觉效果。它主要的原理是通过改变表面光照方程的法线，而不是表面的几何法线，或对每个待渲染的像素在计算照明之前都加上一个从高度图中找到的扰动，来模拟凹凸不平的视觉特征，如褶皱、波浪等等。Blinn 于 1978 年提出了凹凸贴图方法。使用凹凸贴图，是为了给光滑的平面，在不增加顶点的情况下，增加一些凹凸的变化。该方法的原理是通过法向量的变化，来产生光影的变化，从而产生凹凸感。实际上并没有顶点（即 Geometry）的变化。

以下是几种凹凸贴图与其改进方法的总结对比，如图 2.24 所示。

贴图方式	思想概述	提出年代
Bump mapping 凹凸贴图	计算 vertex 的光强时，不是直接使用该 vertex 的原始法向量，而是在原始法向量上加上一个扰动得到修改法向量，经过光强计算，能够产生凹凸不平的表面效果。No self-occlusion, No self-shadow, No silhouette。	1978
Displacement Mapping 移位贴图	直接作用于 vertex，根据 displacement map 中相对应 vertex 的像素值，使 vertex 沿法向移动，产生真正的凹凸表面。	1984
Normal Mapping 法线贴图	normal map 需要法向量的信息，而法向量信息可由 height map 得到，且 texture 的 RGB 可以表示法向量的 XYZ，利用此信息计算光强，产生凹凸阴影的效果。No self-occlusion, No self-shadow, No silhouette。	1996
Parallax Mapping (Virtual Displacement Mapping) 视差贴图	没有修改 vertex 的位置，以视线和 height map 计算较陡峭的视角给 vertex 较多的位移，较平缓的视角给 vertex 较少的位移，透过视差获得更强的立体感，即利用 HeightMap 进行了近似的 Texture Offset。No self-occlusion, No self-shadow。	2001
Relief Mapping (Steep Parallax Mapping) 浮雕贴图	更精确地找出观察者的视线与高度的交点，对应的 texture 坐标则是位移的距离，所以能更正确地模拟立体的效果。Relief Mapping 实现了精确的 Texture Offset。Relief Mapping 可以产生 self-occlusion, self-shadowing, view-motion parallax, and silhouettes。	2005

图 2.24 凹凸贴图与其改进方法的总结对比

表示凹凸效果的另一种方法是使用高度图来修改表面法线的方向。每个单色纹理值代表一个高度，所以在纹理中，白色表示高高度区域，黑色是低高度的区域（反之亦然）。示例如图 2.25。

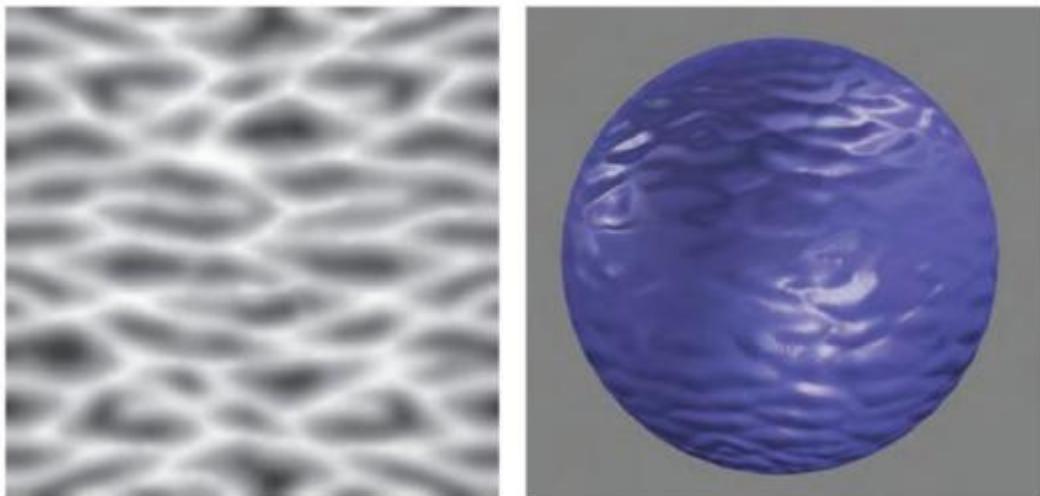


图 2.25 使用高度图来修改表面法线的方向造成凹凸不平的视觉效果

2.3 光学相关技术（反畸变）

2.3.1 光的基本行为

光可以用三种看起来相互矛盾的方式来描述：

1. 光子：在空间中高速移动的微小能量粒子。当考虑传感器或接收器接收的光子数量时，这种解释是有帮助的。
2. 光波：通过空间的波纹，类似于在水面上传播的波浪，但是是三维的。波长是峰值之间的距离。在研究颜色的光谱时，这种解释是有帮助的。
3. 光线：光线追踪单个假想光子的运动。方向垂直于波前（见图 2.26）。这个解释在解释镜头和定义可见性概念时很有用。

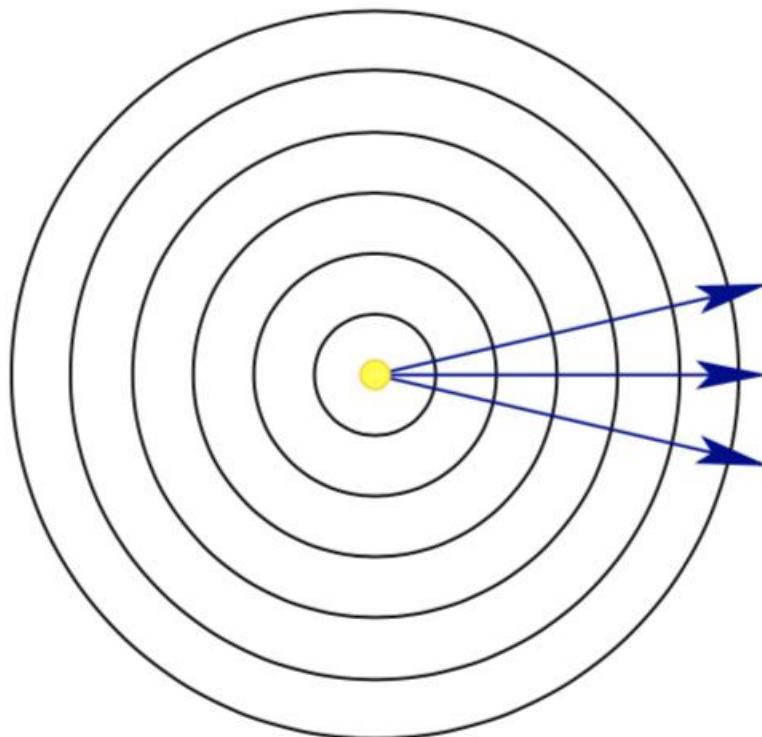


图 2.26 从点光源发出的波和可见光线

与材料的相互作用

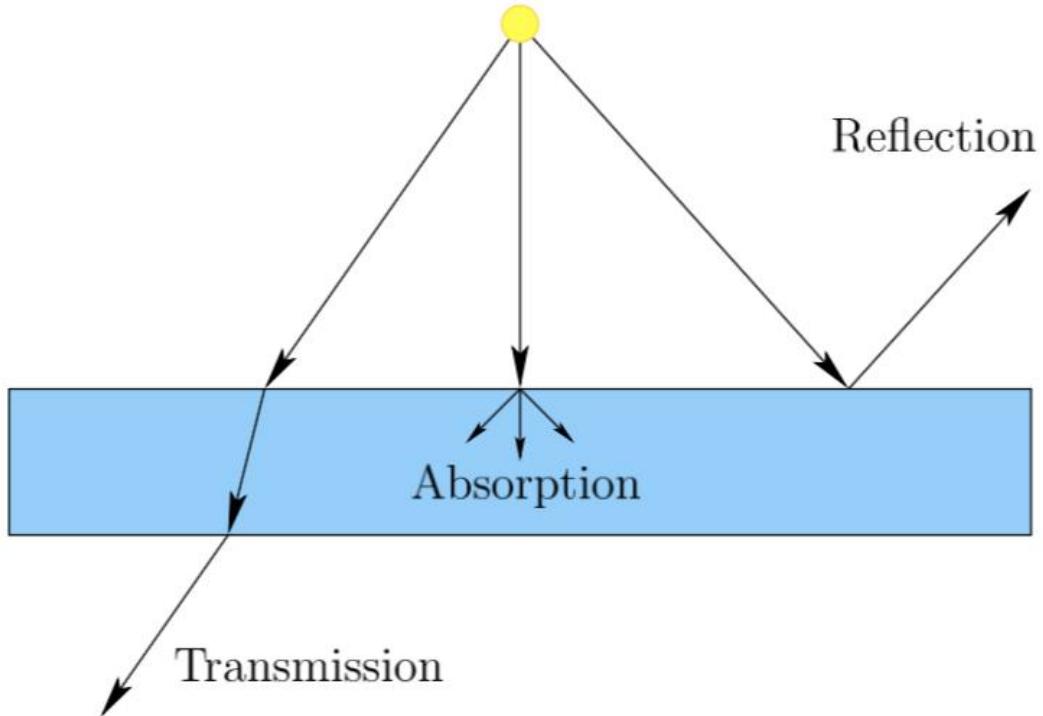


图 2.27 当光能碰到不同介质的边界时，有 3 种可能性：透射，吸收和反射

当光线撞击材料表面时，可能会发生三种行为之一，如图 2.27 所示。在透射的情况下，能量穿过材料并从另一侧离开。对于像玻璃这样的透明材料，透射光线会按照斯涅尔定律减速并弯曲。对于不透明的半透明材料，射线在离开之前会散射到各个方向。在吸收的情况下，当光被捕获时，能量被材料吸收。第三种情况是反射，其中光线从表面偏转。沿着完美光滑或抛光的表面，光线以相同的方式反射：出射角等于入射角。这种情况称为镜面反射，与漫反射相反。反射光线在任意方向上散射。通常，所有三种传播，吸收和反射的情况同时发生。这些情况取决于许多因素，例如接近角度，波长以及两种相邻材料或介质之间的差异。光被不同地模拟为几何光线，电磁波或光子（具有某些波动性的量子粒子）。无论怎样处理，光都是通过空间传播的电磁能。根据渲染目的，光源可以以许多不同的方式来表示。光源可以分为三种不同类型：平行光源、点光源和聚光灯。

2.3.2 光学畸变

沉浸感需要大的视场角，可以通过将一个大的弯曲的球形显示器放到面前的方式来实现，但是这样的方案是非常昂贵的，一个更加实惠的解决方案是通过在一个小的矩形显示屏上增加一个透镜，然后通过透镜来观看显示屏，从而获得更大的视场角，如图 2.28 所示：

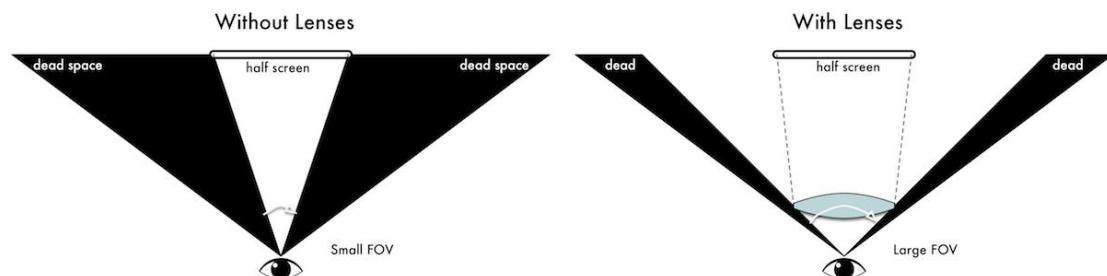


图 2.28 通过添加一个透镜扩大视野范围

虽然放置在眼睛附近的镜片会大大增加视野，但与此同时会付出一定的代价：图像产生

球形失真。视野越大，图像越扭曲。许多统称为畸变的缺陷会降低由镜头形成的图像质量。由于这些问题在日常使用中都很明显，需要采取相关的补偿措施并应用到 VR 系统中。

色差

光通常是一束具有波长光谱的波。当白光通过棱镜折射时，整个可见光谱按颜色被很好地分开。这是一个美丽的光学现象，但对于镜头来说，这是非常糟糕的，因为它分散了图像的各种颜色成分。这个问题被称为色差。

问题的实质在于通过介质的光速取决于波长，图 2.29 展示了简单凸透镜的色差现象。此时，焦距为波长的函数。如果我们沿着相同的光线将红色，绿色和蓝色激光直接照射到镜头中，则每种颜色的光线会在不同的位置穿过光轴，产生红色，绿色和蓝色焦点。对于常见的光源和介质，透过镜头的光线会形成连续的焦点集合。图 2.30 显示了一个具有色差伪像的图像。我们可以通过组合不同介质的凸透镜和凹透镜减少色差，使得发散的射线被强制收敛。

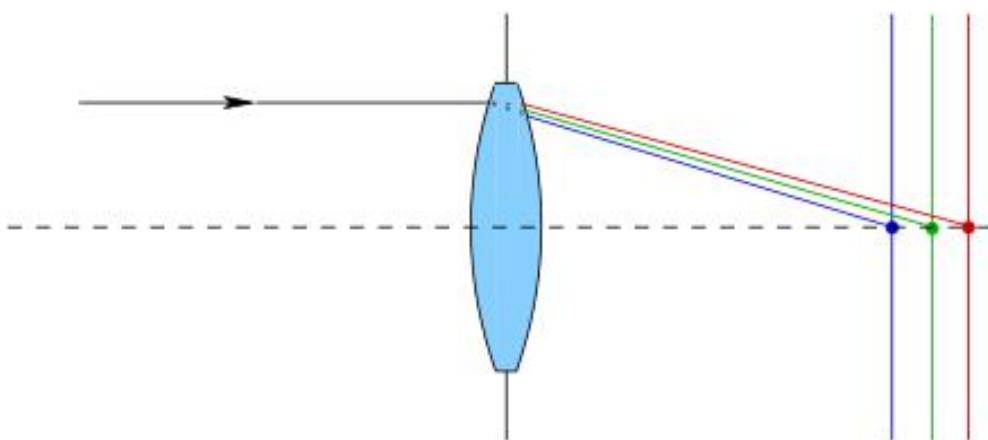


图 2.29 色差是因不同波长的光线在同一介质内的不同光速所引起的，导致每种颜色的光线都有不同的聚焦平面。



图 2.30 上子图被适当地聚焦，而下子图遭受了色差问题。（图由 Stan Zurek 提供）

像散性

图 2.31 描述了像散性，这是对不垂直于晶状体的入射光线发生的晶状体像差。直到现在，我们的镜头图纸都是 2D 的，然而，需要引入第三个维度来了解这种新的畸变。射线可以在一维上发生轴偏移，但在另一维中对齐。通过沿着光轴移动图像平面，不可能使图像成为焦点。相反，这样会出现水平和垂直震源深度，如图 2.31 所示。

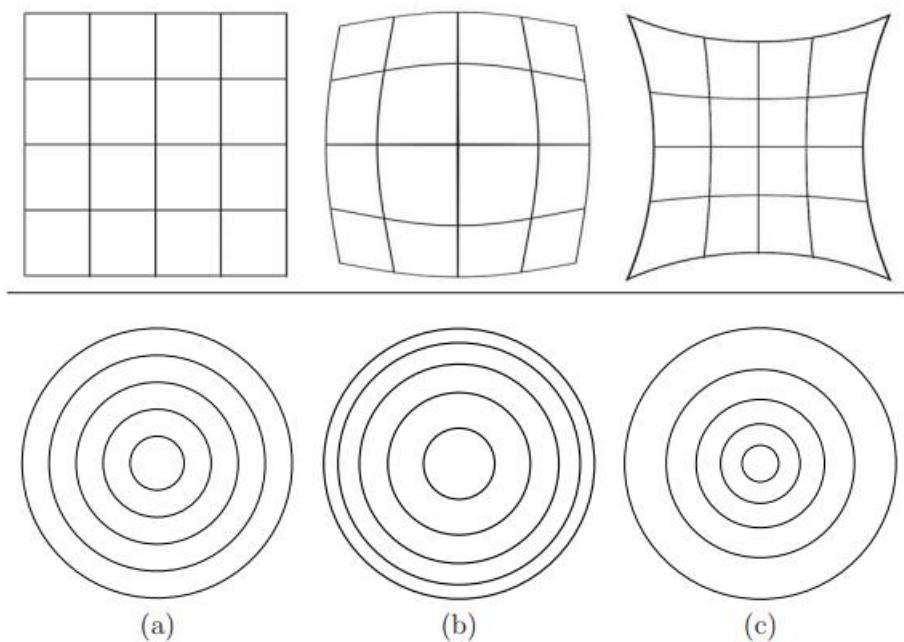


图 2.31 常见的光学畸变。（a）原始图像（b）桶形失真（c）枕形失真。对于第一行的畸变，网格变得非线性扭曲。第二行说明它仍然保持圆对称。



图 2.32 一个带有桶形失真的图像，由鱼眼镜头拍摄。（图片由维基百科用户 Ilveon 提供）

光学畸变的修正

对于大视场的光学系统，桶形畸变和枕形畸变是很常见的（如图 2.32、2.33）。通过 VR 头显镜头观看时，通常会产生枕型畸变。如果图像不经过任何修正的话，那么虚拟世界看上去就会出现扭曲的现象。如果用户的头部来回转动的话，由于四周的变形比中心强烈，一些固定线条（如墙壁）的曲率会动态的改变。如果不加以修正，就没有一种静态物体的感觉，因为静态物体不应该会有动态变形。此外，这也有助于研究导致虚拟现实疾病的成因，可能是由于在 VR 体验时感受到了四周异常的加速度。

那么，如何解决这个问题呢？现如今已经有很多相关的研究，可行的解决方案包括许多不同的光学系统及显示技术。例如，数字光处理（DLP）技术在不使用透镜的情况下可以直接将光投射到眼睛中。

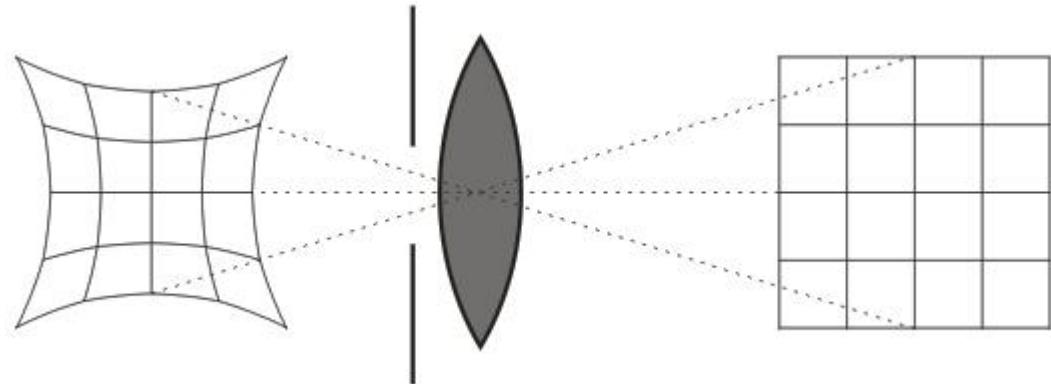


图 2.33 头戴式显示器的镜头枕形失真

解决方法是对这些畸变的图像使用“桶形”畸变，如图 2.34 所示，当我们通过畸变透镜上看，这些经过反畸变的图像看起来就是正常的：

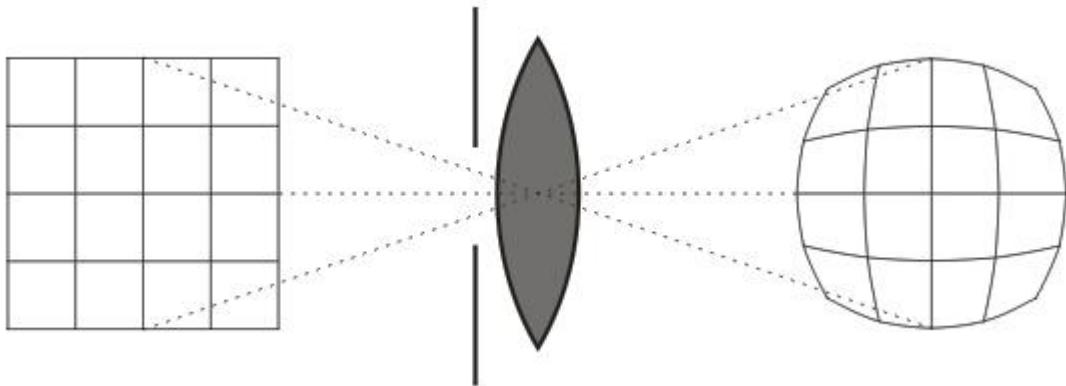


图 2.34 对图像提前进行桶形畸变，使得图像正常

无论畸变的严重程度有多大，都可以通过软件进行修正。假设默认畸变是循环对称的，这意味着畸变量仅取决于到镜头中心的距离，而不取决于中心的特定方向。即使镜头的畸变是标准的圆形对称，它也必须要放置在眼睛中央。一些头戴设备支持 IPD 调节，可以调节镜头之间的距离，使其可以匹配用户的眼睛。如果眼睛不在镜头中央，则会出现不对称畸变。这种情况并不能视为完美对称，因为随着眼睛的转动，瞳孔也会沿着球形弧面移动。随着镜头上瞳孔位置的横向变化，畸变会变得不对称。这促使厂家使用尽可能大的镜头来避免这种问题。另一个原因是，随着镜头和屏幕之间距离的变化，畸变也会变化。这种调整可以满足近视或远视用户的需求，正如三星 Gear VR 头显所做的。这种调整在双筒望远镜中也很常见，这就解释了为什么很多人在使用时不需要戴眼镜。为了正确地处理畸变问题，头戴设备应当能够准确地检测调整设置，并将这些因素考虑在内。

基于像素的处理

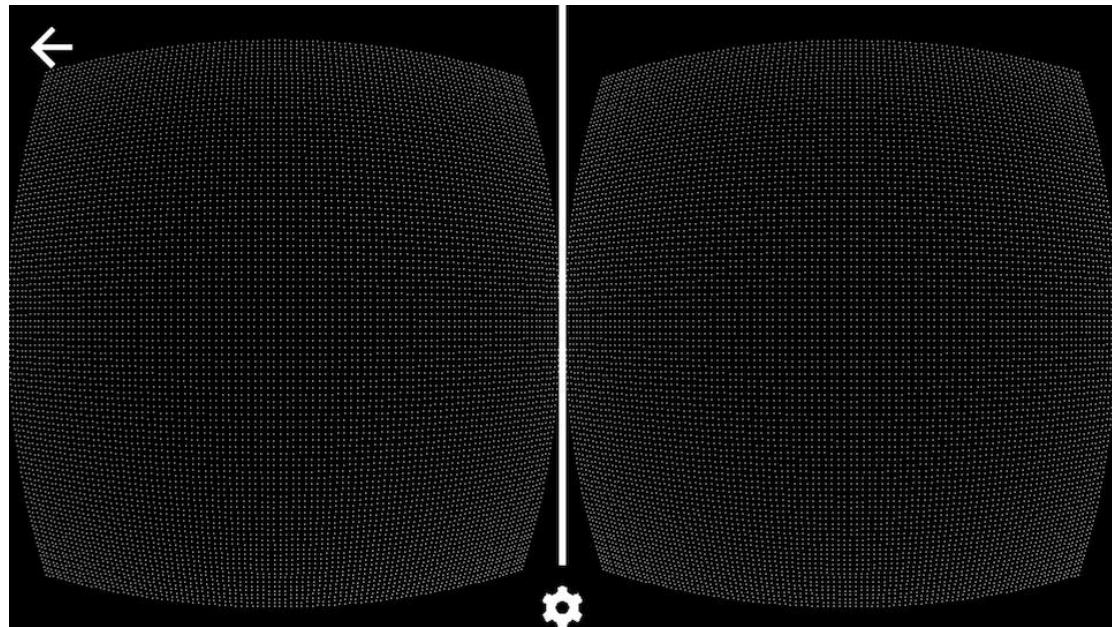


图 2.35 对所有像素进行处理

透镜失真在数学上是很好理解的，由方程控制，失真系数对应于特定透镜。要正确地消除失真，我们还需要计算眼睛的中心，这就需要了解显示器的几何形状和外壳本身。在此方法中我们单独处理每个像素，对每个像素进行“桶型”畸变的数学变换。首先确定特定头显的径向畸变函数 f ，将特定镜头置于屏幕前固定距离的位置。这是一个回归或曲线拟合问题，步骤包含测量许多点的畸变从而确定参数等，然后取最佳的拟合。其次确定 f 的逆函数，

使得可以在镜头产生畸变前将其应用到图像渲染上。 f 的逆函数可以抵消畸变带来的影响。然而，多项式函数通常没有确定的或是闭合形式的逆函数，因此经常使用近似的函数进行拟合。

基于网格的处理

与基于像素的处理方法不同，这里并不是单独处理每个像素，而是扭曲相对稀疏网格的顶点，如图 2.36 所示。

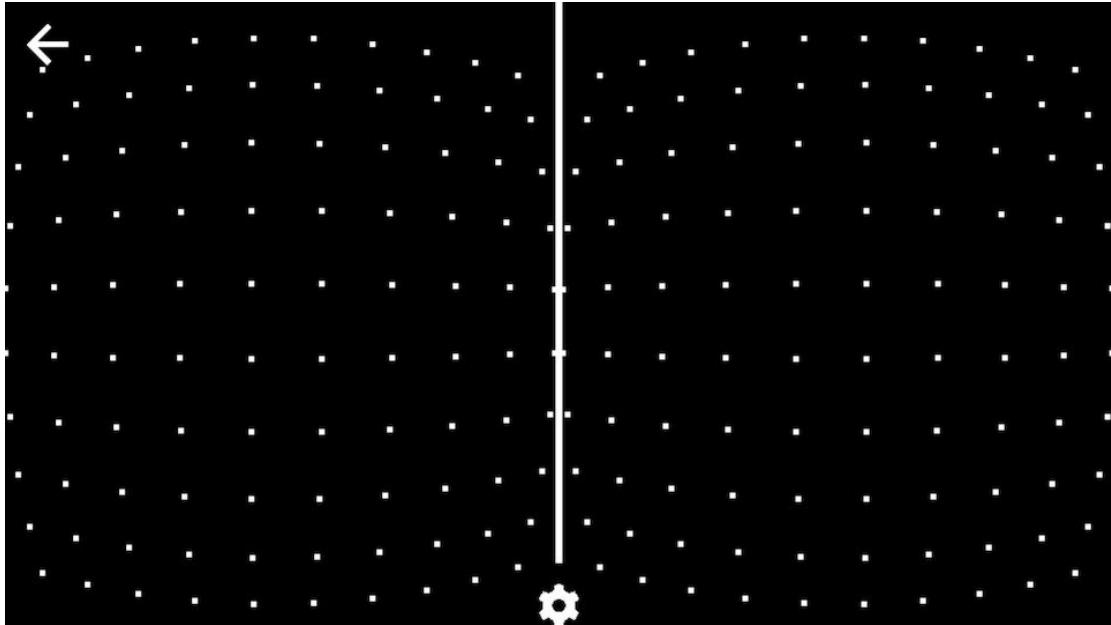


图 2.36 基于网格的处理

对网格中的每个顶点进行计算可以节省一些直接计算量，计算步骤显著减少，并且性能得到了很好的提升。

2.3.3 异步时间扭曲

异步时间扭曲 (Asynchronous Timewarp 简称 ATW) 是一种生成中间帧的技术，当游戏不能保持足够帧率时，ATW 能产生中间帧，从而有效减少游戏画面的抖动。

大部分的显示器和绝大部分的手机屏幕的刷新率都是 60Hz，也就是说，在理想情况下我们的显示设备大概要在每秒处理 60 帧的画面，也就是说从数据到渲染就有 $1000/60 \approx 16.6666\text{ms}$ 的时间延迟。对于虚拟现实设备，为了要在虚拟世界里呈现给人们正确的感知图像，必须要在显示器上定时更新图像，然而，如果渲染时间太长，对应帧就会丢失，产生的结果就是抖动，帧率不稳定。那么，如何抵消这个时延呢？John Carmack 提出一种方法：通过大量采集陀螺仪数据，在样本足够多的情况下，就可以预测出 16.67ms 后用户头部应有的旋转和位置，按照这个预测的数据来渲染。时间扭曲则是一种图像帧修正的技术，它通过扭曲一幅将被送往显示器的图像，来解决这个 16.67ms 的延迟。最基础的时间扭曲是基于方向的扭曲，这种只纠正了头部的转动变化姿势，这种扭曲对于 2D 图像是有优势的，它可以合并一幅变形图像且不需要花费太多系统资源。

异步时间扭曲是指在一个线程（称为 ATW 线程）中进行处理，这个线程和渲染线程平行运行（异步），如果没有时间扭曲，HMD 将捕获有关头部位置的数据，根据此数据渲染图像（正确的角度等），然后在下一个场景到达屏幕时显示图像。在 60 fps 的游戏中，每 16.7 毫秒显示一个新场景。通过此过程，您看到的每个图像都基于近 17 毫秒前的头部跟踪数据。使用时间扭曲，该过程的前两部分是相同的。HMD 将捕获有关头部位置的数据，并根据数据

渲染图像。在显示此图像之前，HMD 会再次捕获头部位置。使用此信息修改渲染图像以适合最新数据。最后，修改后的图像显示在屏幕上。生成的图像更准确地描绘了显示时头部的位置，而不同于最初渲染的图像。时间扭曲仅适用于非常短的时间间隔。

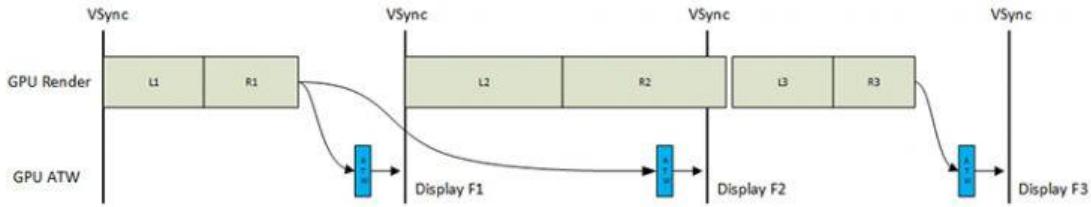


图 2.37 GPU 渲染过程中插入 ATW

GPU 分别为左右眼的画面进行渲染，然后在画面显示出来之前插入一个 ATW 的处理过程。在左边这帧的处理中，画面渲染及时完成，此时进行直接显示，若中间的第二帧渲染未能及时完成，会出现画面抖动。但添加的 ATW 进程会将前面一帧调用出来重新显示，同时加上头盔运动变化，从而保持帧率。

异步时间扭曲局限性

1： 它需要 GPU 硬件支持合理的抢占粒度。在 90Hz 时，帧间间隔大约是 11ms，这意味着为了使 ATW 有机地生成一帧，它必须能够抢占渲染线程并且运行时间少于 11ms，然而 11ms 并不友好，如果 ATW 在一帧时间区间内任意随机点开始运行，那么潜伏期（执行和帧扫描之间的时间）也将随机，并且需要确保不跳跃任何渲染帧。而对现在的图形驱动实现来说，2ms 抢占是一个艰巨的任务。

2： 它要求操作系统和驱动程序支持 GPU 抢占。如果抢占操作不是很快，则 ATW 将无法抢在画面同步之前生成中间帧。这样会使得最后一帧再显示，进而导致抖动，这意味着正确的异步时间扭曲实现应该能够抢占和恢复任意渲染操作和管线状态。理论上讲，目前已有的三角抢占（triangle-granularity）也不够好，因为我们不知道一个复杂着色器执行将花多长时间。

另一方面是操作系统对抢占的支持，在 Windows 8 之前，Windows 显示驱动模型（WDDM）支持使用“批处理队列”粒度的有限抢占。但不幸的是，图形驱动程序趋向于大批量渲染效率，会导致 ATW 的实现过于粗糙。

总体来说 ATW 确实是一项很棒的技术，没有它的话，开发者在游戏开发中为了保持画面帧率只能非常保守地使用 CPU 和 GPU 性能，而 ATW 可以更容易地保持帧率稳定，从而让开发者在画面设计上更加大胆。实际运行 Oculus 中可以发现，没有使用 ATW 的 app 在运行中丢失了约 5% 的帧。ATW 可以将大部分丢失的帧补上，从而大幅减少画面抖动。而这一切对 app 来说不需要消耗更多性能或更改代码就能实现。Oculus 还表示这一切只是开始，他们正与合作伙伴尝试提高 ATW 的运行效率。

2.4 常用的终端集成引擎

对于使用者来说，关于 VR 技术的最直接感受就是全景视频，也称 360 度全景视频或沉浸式视频。现在有很多终端集成引擎能够实现全景视频的呈现与播放，就目前来说，使用最多的是 Unity3D 软件，Unreal 和 WebVR 等，这些都能够很好的完成全景视频的呈现。

Unity3D

Unity3D 软件是由 Unity Technologies 开发的一个能够创建三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的游戏和相关软件的开发工具。该工具可以发布

游戏或者软件至 Windows、Mac、Wii、iPhone、WebGL(需要 HTML5)、Windows phone 和 Android 平台，也可以利用 Unity web player 插件发布网页游戏，支持 Mac 和 Windows 的网页浏览，它的网页播放器也被 Mac 所支持。

下图是 Unity 的初始界面：

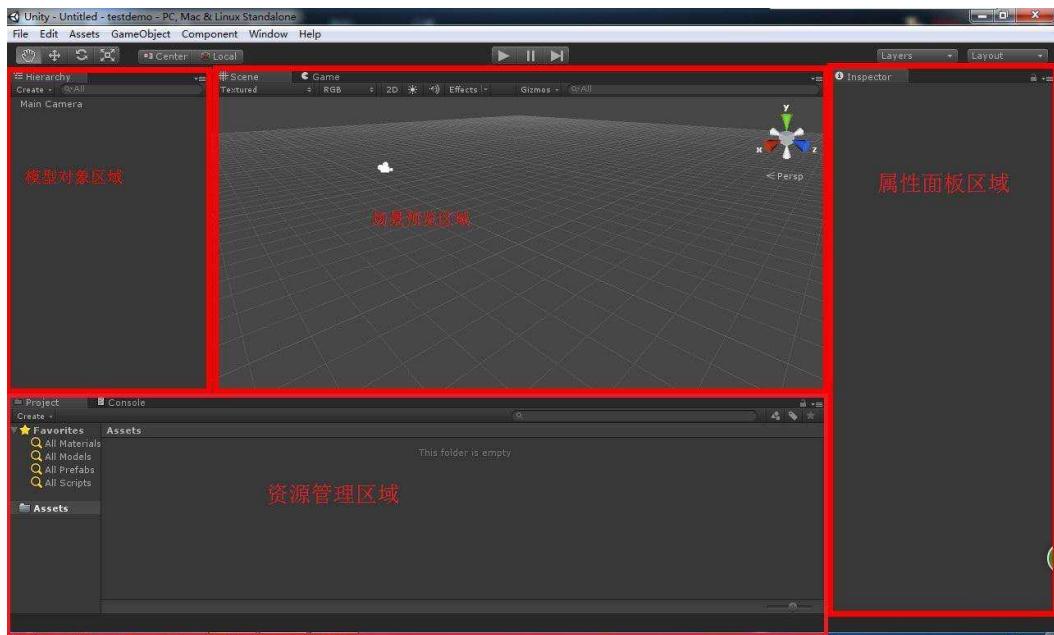


图 2.38 Unity 初始界面

要想很快地完成一个能够实现全景视频呈现的软件，一个比较好的方式就是通过使用现有 Unity 中的 SDK，比如 googlvr for unity，Oculus Unity SDK，AVpro SDK 等，这些 SDK 中都很好的集成了相关功能，方便了开发者的使用。下面介绍如何通过 Unity 和 GoogleVR for Unity 来创建一个简单的全景视频播放器。

首先新建一个 unity 项目，在新建场景中添加一个球体，position 设置为原点(0,0,0)，size 可以设置为(5,5,5)，也可以根据需要设置大小，MainCamera 的 position 同样设置为原点(0,0,0)，这样就相当于观看者站在球心的位置，而相应的球体内部应该播放的是全景视频，Unity 默认是不会将球体的内部渲染出来，所以现在需要通过着色器翻转球体的法线。

新建材质给球体，再将新建的 shader 给新建的材质。Shader 代码如下：

```

1 Shader "Custom/flipnormal" {
2     Properties {
3         _MainTex ("Albedo (RGB)", 2D) = "white" {}
4     }
5
6     SubShader {
7         Tags { "RenderType"="Opaque" }
8         Cull Off
9         CGPROGRAM
10        #pragma surface surf Lambert vertex:vert
11        sampler2D _MainTex;
12        struct Input {
13            float2 uv_MainTex;
14            float4 color:COLOR;
15        };
16
17        void vert(inout appdata_full v)
18        {
19            v.normal.xyz=v.normal*-1;
20        }
21        void surf (Input IN, inout SurfaceOutput o) {
22            fixed3 c = tex2D (_MainTex, IN.uv_MainTex);
23            o.Albedo = c.rgb;
24            o.Alpha = 1;
25        }
26        ENDCG
27    }
28    FallBack "Diffuse"
29 }

```

接下来为球体添加 VideoPlayer 组件，将准备好的.mp4 全景视频拖至 Video clip

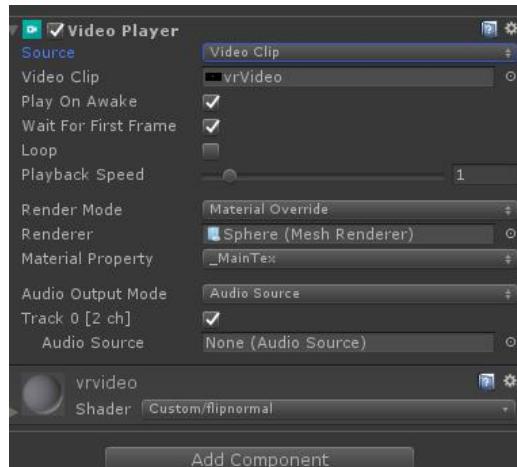


图 2.39 Video Player 组件

将 GoogleVR SDK 导入，并更改一些相关的设置：依次选择 Unity 菜单栏中的 File-Build Settings，将当前的场景添加进列表中，选择 Android 作为输出构建平台。在将平台切换完成之后，点击 Player Setting 打开播放器设置，将 Other Settings 下的 Virtual Reality Supported 勾选，并点击下面的加号，选中 Cardboard 添加至列表。

在将上述设置都完成后，将 GoogleVR/Prefabs 文件夹下的 GvrViewerMain 预制件拖拽到场景中，在 inspector 中将坐标设置为球体中心(0, 0, 0)。最后导出 APK 到 Android 设备上就可以在手机端观看视频了。GoogleVR 的其他功能用户也可以动手实现，并且现阶段的许多大厂商都已经在研发相关 VR 全景视频播放的内容，许多 unity 的 SDK 可以用来实现全景视频的播放并且开发者使用起来也十分方便。

Unreal Engine 4

UE(Unreal Engine)是全球顶尖游戏引擎，占用全球商用游戏引擎 80%的市场份额。虚拟引擎是美国 Epic 游戏公司研发的一款高水准游戏引擎，渲染效果强大，采用了 pbr 物理材质系统，不仅能够用于制作主机游戏，PC 游戏，手机游戏，还能够涉及高精度模拟，可视化与设计表现，无人机巡航等诸多领域。并且 UE4 为手柄、VR 控制器提供了良好支持。下面介绍如何在 UE4 中播放全景视频。

首先在 Content Browser 中展开 Sources Panel，并且在 Content 下创建名为 Movies 的文件夹，如下图：

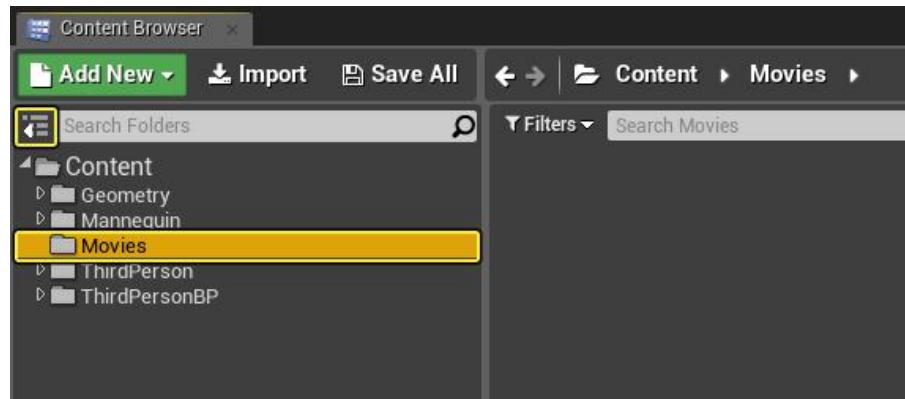


图 2.40 Content Browser

在 Movies 文件夹上点击鼠标右键，在出现的菜单中选择 Show in Explore。接下来将全景视频文件或者其他格式的视频文件拖到 Content/Movies 文件夹下，好能够保证视频能够被正确打包。

在 UE4 的项目工程中，在 Movies 文件夹上右键鼠标点击，在 Media 下选择 File Movie Source：

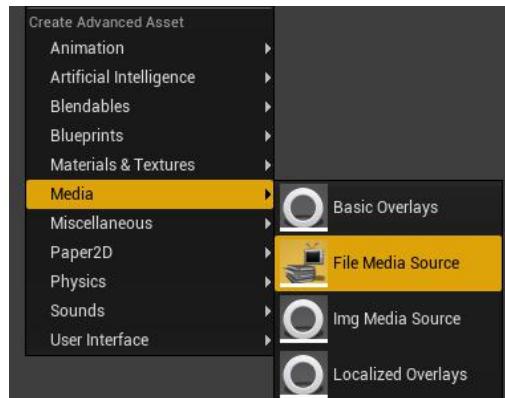


图 2.41 选择文件来源

将这个资源命名为 SampleVideo，打开以后，在 File Path 处将其定位到 Content/Movies 文件夹中的视频：

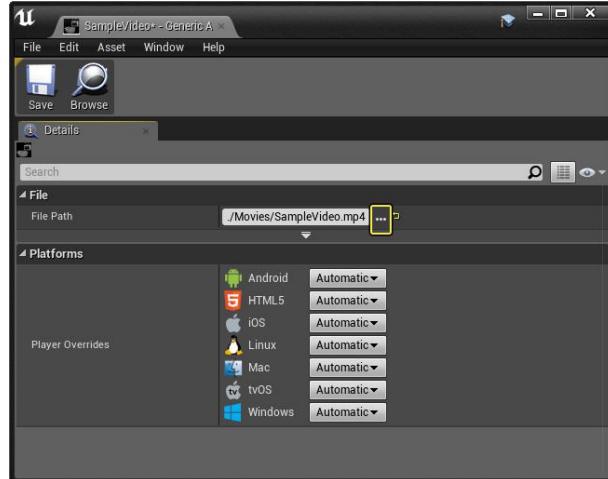


图 2.42 定位文件

接下来在 Content Browser 出点击鼠标右键，然后在 Media 下选择 Media Player 资源：

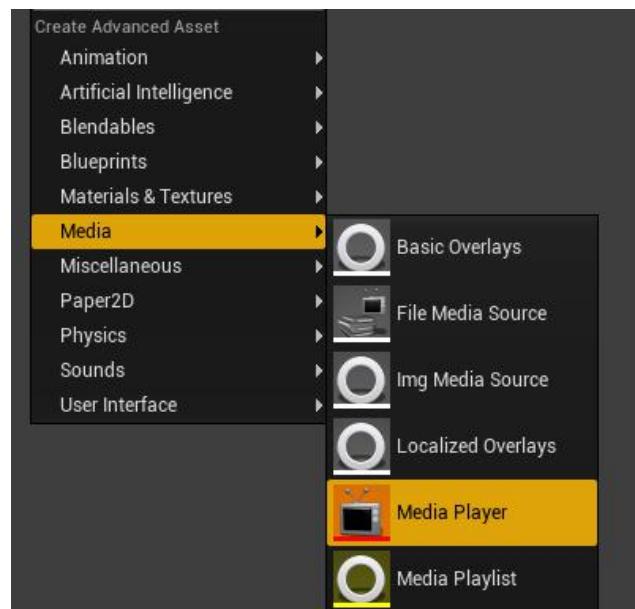


图 2.43 Media Player

在弹出的 Create Media Player 窗口中，点击选择 Audio output SoundWave asset 和 Video output Media Texture asset。通过选择这两项，会自动创建 SoundWave 和 MediaTexture 资源，并且会与播放视频所需的 MediaPlayer 资源相关联。

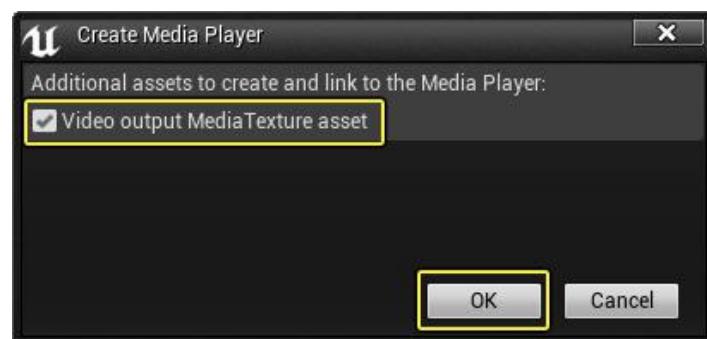


图 2.44 关联 asset

接下来对新的 MediaPlayer 资源命名，这里将其命名为 MyPlayer，相对应的 SoundWave

和 MediaTexture 也会改变；打开 Media Player 资源，双击 Media Source 资源，然后视频就会播放，并且在右下角的 Details 面板里，Output 部分 SoundWave 和 Video Texture 会被自动赋值。

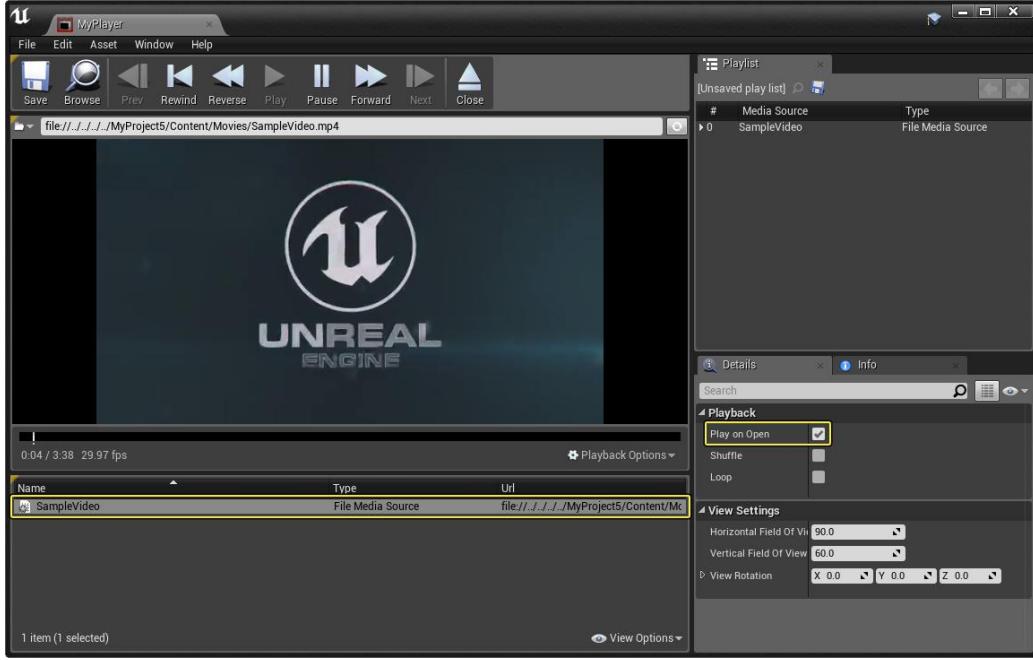


图 2.45 命名资源

按住 Ctrl 然后同时选择 SoundWave 和 Media Texture 资源，将其拖放到创建的球体 Mesh 上（自己提前创建一个球体即可），这将自动创建 Material 并将其赋予到 Static Mesh 上。接下来再点击工具栏中的 Blueprints 按钮然后点击 Open Level Blueprint：

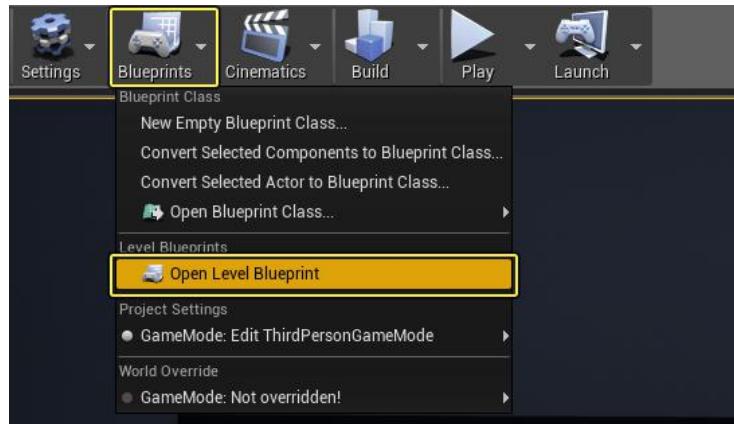


图 2.46 Open Level Blueprint

添加一个 Media Player Reference 类型的 Variable，并且将其命名为 MediaPlayer，而且将其设置为 MyPlayer 的 MedaiPlayer 资源。然后需要 Compile 这个资源，才能够设置 Default Value。

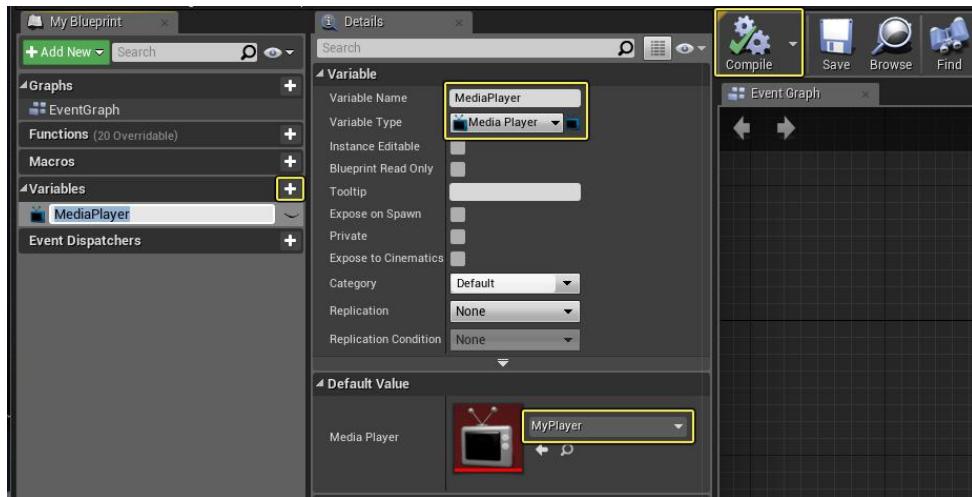


图 2.47 编译资源

然后按住 Ctrl 并将变量 MediaPlayer 拖放到 Event Graph 窗口，然后点击鼠标右键并添加一个 Event Begin Play 节点：

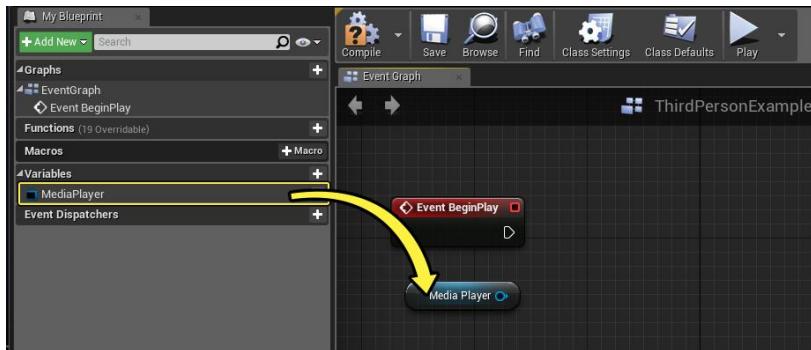


图 2.48 添加节点

拖动变量 MediaPlayer，然后使用 OpenSource 节点，将其 MediaSource 设置为 SampleVideo，如下图所示：

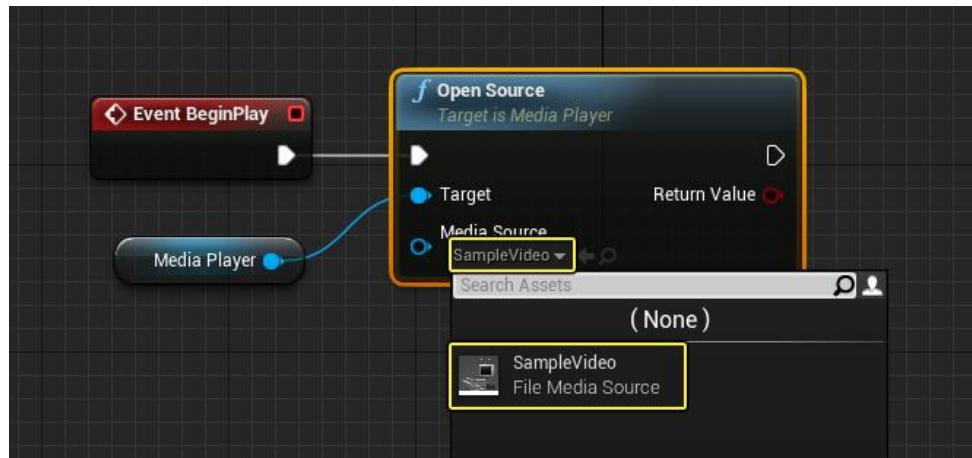


图 2.49 设置 Sample Video

最后，关闭 Level Blueprint，然后在编辑器内点击 Play 按钮即可。

根据以上介绍的过程来看，使用 UE4 来播放全景视频原理上是和 Unity 一样的，但是实际操作起来的难度却有所增加，而且相对 Unity 来说，UE4 的成本更高，使用难度更大，需要更多的时间去熟练相关的操作。

WebVR

WebVR 顾名思义，就是 web+VR 的制作方式，使用户能够在网页上观看全景视频或者全景图片。WebVR 有两种体验方式，一种是 VR 模式，另一种是裸眼模式。VR 模式下，可以在移动设备上观看，比如 Cardboard 等来体验手机浏览器的 WebVR 网页，而浏览器会根据陀螺仪的参数等来获取用户的头部的倾斜角度和转动的方向进一步告知页面需要渲染哪一个朝向的场景；还可以在 PC 端观看，比如通过佩戴 Oculus Rift 的分离式头显浏览连接在 PC 主机端的网页，现在 WebVR 还没有广泛应用，支持 WebVR API 的浏览器主要是火狐的 Firefox Nightly 和设置了 VR enabled 的谷歌 Chrome beta。还有就是裸眼模式，在 PC 端，裸眼模式应该允许用户可以使用鼠标拖拽场景，实现视角的转动；在移动端，则应让用户使用 touchmove 或旋转倾斜手机的方式来改变场景视角。

需要使用到的框架有 Three.js，它是构建 3D 场景的框架，封装了 WebGL 函数，简化了创建场景的代码成本，需要引入的插件有 three.min.js 和 webvr-polyfill.js，后者提供了大量 VR 相关的 API，比如 Navigator.getVRDisplay() 获取头显信息的方法。现在已经有很多能够支持全景视频播放的网站，比如有 YouTube、steam 等平台，国内的优酷，可以定制的 play2VR 等等，这些都可以很好地播放全景视频。

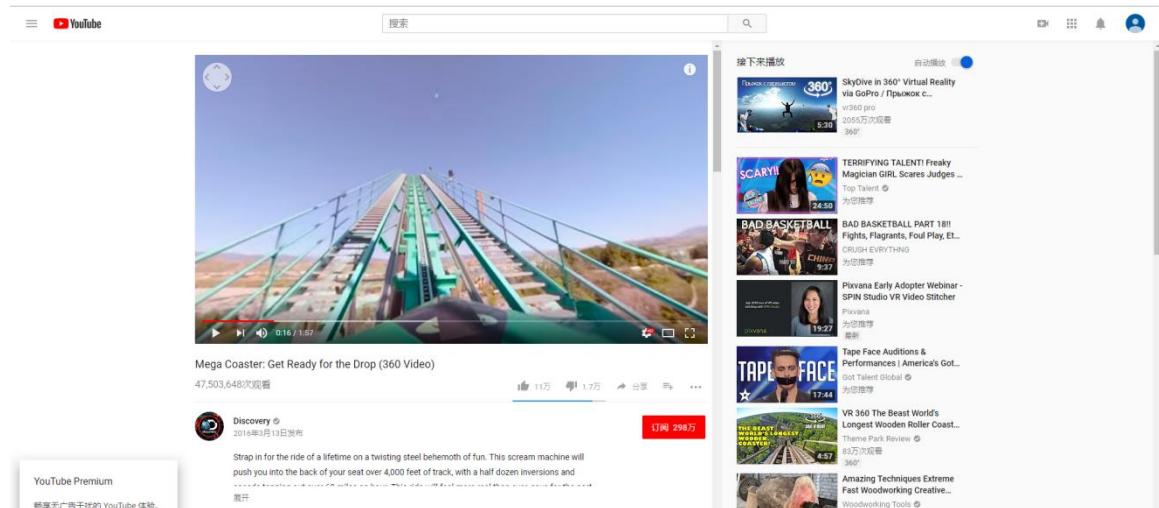


图 2.50 YouTube 中的全景视频

就像 YouTube 网站一样，当把鼠标移动到视频上面时，会出现能够拖动视角的手性标志，进而点击鼠标左键拖动即可。

现阶段各大厂商均已加入到了全景视频的发展之中。在可预见的未来，VR 将会发展迅猛，是未来科技发展的重头戏，全景视频的呈现也将会变得越来越简单和普及，人们将会更容易地享受到观看全景视频的乐趣。

2.5 音频相关技术

2.5.1 声场

传统的音频信号一般使用麦克风进行捕获和记录，而最终通过扬声器等设备回放声音信息。这是目前仍比较常用的一种做法，但是这种方法的前提是声音对象可以通过点源捕获，然后由一个（或少量）点源渲染便可充分表示。

声场表示则消除了这种假设，允许捕获和回放完整的音频声场，用户就像在现实世界中体验一样。

2.5.2 声场通信场景

以下是涉及声场捕获和重建的多种场景：

1. 声场捕获和记录：以尽可能高的保真度捕获和记录声场。没有带宽限制且没有传输信道，但可能需要压缩来缓和数据量。传感器的数量将影响相应情况下的要求。
2. 声场重放和重建：这是捕获和记录的逆过程，这种情况下声场是通过不受带宽限制的媒体播放的。
3. 声场单向广播/流传输：这是一对多场景，该情况下声场通过带宽限制的信道传输，最终传至多位置的终端。
4. 声场双向通信：这种情况设想了双向通信的情形，其中任意一端都具备发送和接收声场的功能。

2.5.3 声场传感器配置

考虑上述所有情况的两个关键的方面：声场捕获和声场重建。

“平面”上的传感器

声场捕获和重建的简化解释是想象听众和感兴趣的声音对象之间的无限平面，当声波从物体传播到听者时，它们穿过平面。如下图所示：

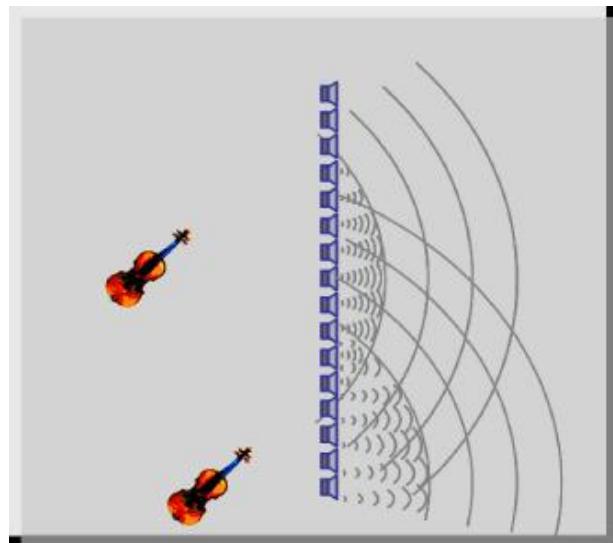


图 2.51 穿过平面的波场

传感器在平面中的最有效放置是六边形填充，如下面的图 2.52 所示，其中“X”表示传感器位置。

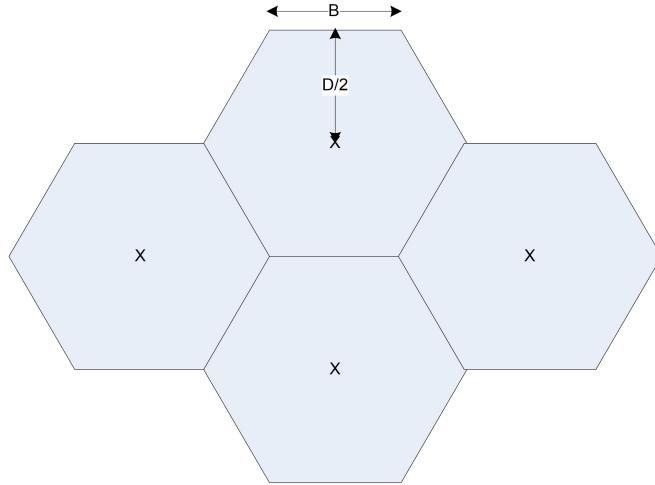


图 2.52 具有六边形拼接的传感器

虽然人类认知的真实世界是模拟的，但研究人员总希望将这个世界各个方面信息表示为一些数字序列。对于单个音频传感器，我们可以假设 48kHz 的（时间）采样频率 f_s 足以表示音频信号，因为人们普遍认为人类听觉的频率范围是 20Hz 到 20kHz。

声场的捕获同时需要最小的时间采样频率和最小的空间采样频率。空间采样频率由传感器的空间位置确定。假设传感器使用六边形平铺均匀分布在平面上，如上图 2.52 所示。任意一个传感器到其最近传感器的距离 D 为：

$$B^2 = (D/2)^2 + (B/2)^2 \quad (2.1)$$

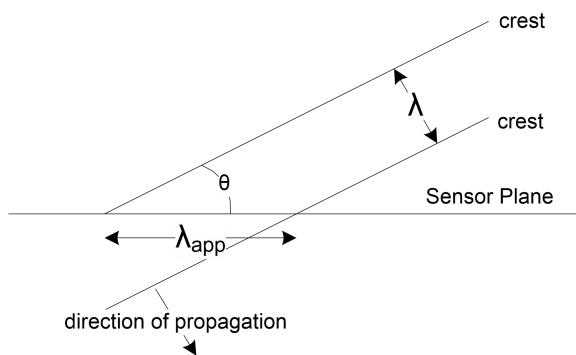
或

$$D = B\sqrt{3} \quad (2.2)$$

其中 B 是六边形边缘的长度， $D/2$ 是从六边形中心到边缘中点的距离。

一个更简单的分布是仅考虑水平线而不是平面上的传感器。在这种情况下，传感器将以距离 D 分开放置。

在平面或线性配置中，如果传感器被间隔得太远，则重建的声场将遭受“空间混叠”，因为它不能精确地捕获波长比临界距离 f_c 更短（频率更高）的信号分量。这种情况在下面的图 2.53 中得以说明。该图显示出了以角度 θ 照射在传感器平面上的平面波。波长是 λ ，波峰逐个测量。

图 2.53 以角度 θ 撞击传感器平面的平面波

由于入射角的存在，平面上的传感器感应到更长的波长 λ_{app} ，即：

$$\lambda_{app} = \frac{\lambda}{\sin \theta} \quad (2.3)$$

入射角决定了临界频率：

$$f_c = \frac{c}{2D \sin \theta} \quad (2.4)$$

或

$$D = \frac{c}{2f_c \sin \theta} \quad (2.5)$$

其中 c 是声速， D 是传感器间距。临界频率 f_c 是传感器可以在没有混叠失真的情况下检测到的最大频率。随着入射角 θ 增加，当 $\theta = 90$ 度时，临界频率降低到极限：

$$f_c = \frac{c}{2D} \quad (2.6)$$

虽然我们假设了一个无限的传感器平面（或线），但这在实际中是不可能的。有限范围的传感器线阵或平面在声场捕获和重建中会引起“截断效应”失真。

在信号处理术语中，这是空间域中的频谱泄漏，并且是由于以矩形函数作为窗口函数所引起的。如果外部传感器或转换器的检测或再现水平降低，则可以减少泄漏程度。这对应于信号处理中使用边缘逐渐变细的窗口函数。

球面上的传感器

平面传感器的一种变体是球面传感器。当球体的半径 R 变为无穷大时，球面就变成一个平面，因此有很多共同之处。然而，球面传感器的概念存在更吸引人的方面，这里说明两种不同的场景。

第一种情况是以用户为中心，捕获影响虚拟用户的声场，然后在不同的时间和地点为实际用户再现声场。在这种情况下，声场由距离虚拟用户 R 的物体产生。

第二种情况是以发声物体为中心，捕获从实际声音对象发出的声场，然后在不同的时间和地点再现以模拟虚拟对象。在这种情况下，声场由实际声音物体产生并且被用户感知到一系列远离自身虚拟对象的发声物体。

在每种情况下，我们可以认为声场是由半径为 R 的球体表面上的传感器捕获的。在以用户为中心的情况下，传感器（即麦克风）面向外，或者其方向性图案朝外，以捕获球体之外的声音。在以物体为中心的情况下，传感器面向内，或者其方向性图案面向内，以捕获球体内部的声音。如果捕获传感器具有全指向性，则两种情况下的传感器配置是相同的。

对于声场重建，以用户为中心的情况，球体表面上具有转换器（即扬声器），其具有面向内的方向性图案，而以物体为中心的壳体将具有面向外的方向性的转换器。

下面的图 2.54 显示了以用户为中心或以物体为中心的情况的传感器/传感器配置。

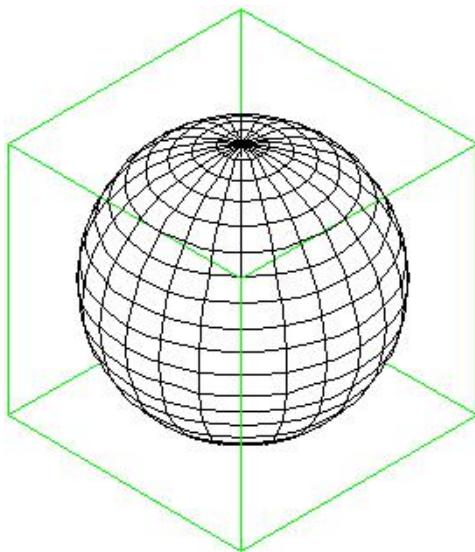


图 2.54 封闭球体上的传感器。声源对象位于球体的中心，用户位于外部，或者用户位于球体的中心，源对象位于外部。

如果值 R 足够大，则可以为单个，少量甚至大量个体定制以用户为中心的场景。如果 R 足够大，则可以为单个物体或多个物体定制以物体为中心的情况。或者，为了达到声场捕获的目的，每个发声物体可以拥有一个球体，即每个发声对象可以有类似的单个球体用于重建。

如果半径为 R 的球体的表面均匀地铺设有六边形，那么传感器的数量 N 是球体的表面除以六边形的面积。使用图 2.52 中的尺寸 D 和 B ，以及上面的临界频率 f_c 的等式，则 N 为：

$$N = \frac{4\pi rR^2}{(B^2)(3\sqrt{3}/2)} = \frac{4\pi R^2}{D^2(\sqrt{3}/2)} = \left(\frac{2}{\sqrt{3}}\right) \frac{4\pi R^2}{\left(\frac{c}{2f_c \sin \theta}\right)^2} \quad (2.7)$$

可以看出，关于最大入射角的假设对于确定所需传感器的数量是至关重要的。但是，如果需要高达 60° 的入射角，那么半径为 4 米的球体将需要 $N = 600\,000$ 个传感器才能完全捕获具有 20 至 20kHz 带宽的声场。

2.5.4 沉浸式媒体中的音频

我们所处的现实环境中声音来自四面八方，因此对于周围的环境状况和发生的事能够产生直接、准确的判断，在沉浸式的虚拟环境中，同样需要让用户听到来自四面八方的声音，才有助于在虚拟环境中产生真正的沉浸感。

关于如何在 VR 中实现这一点，首先来看我们在日常的 3D 电影、3D 游戏中已经接触到的 3D 音效。看 3D 电影的时候，由于声源有确定的空间位置，声音有确定的方向来源，因此人们能够辨别到声源的方位。5.1 环绕声的效果比双声道立体声更加“3D”，7.1 环绕声比 5.1 环绕声更加“3D”，在一定上限范围内，音箱数量越多，3D 环绕声系统的效果就越好。但这样的环绕声系统的音箱位置是在同一平面上，因此现在又有了“杜比全景声”，在影厅的天花板上也装有音箱，这样观众就能听到来自头顶的声音，与环绕声是一样的设计思路。

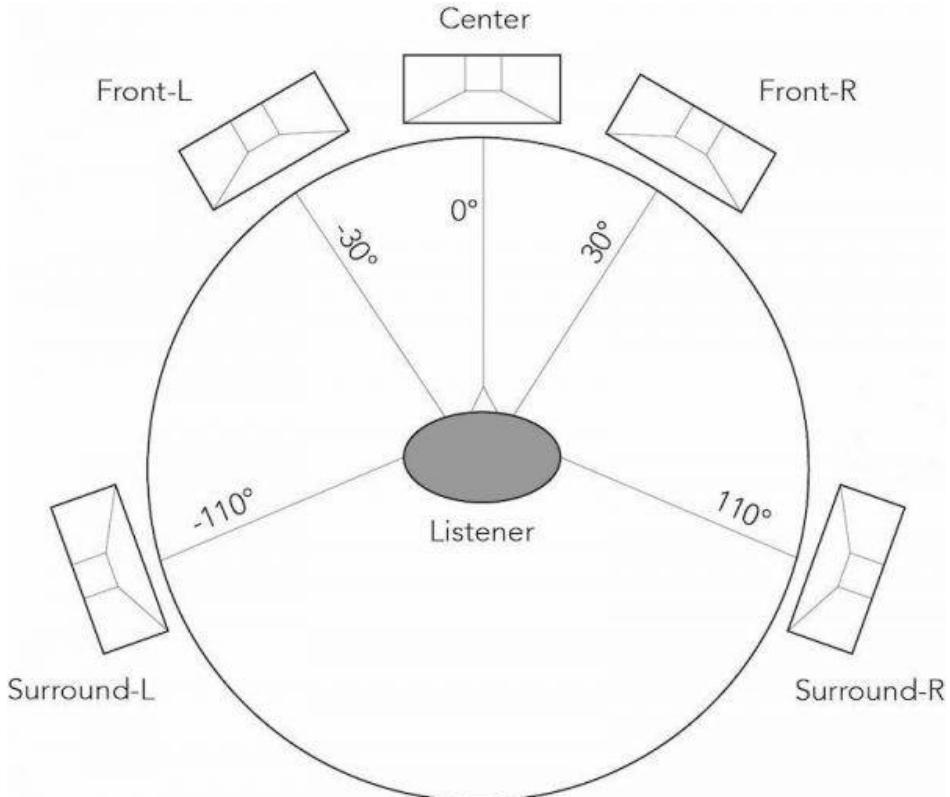


图 2.55 5.1 环绕声

而在 VR 中，观众处于场景中心，可以自主选择观看的方向和角度，用户要通过头显加耳机的方式感受 VR 体验，就需要在双声道立体声输出的耳机上听到来自各个方向的声音。

另一方面用户需要来回转动头部或者有大幅度的身体运动，因此还要考虑身体结构对于声音的影响。因此在 VR 中需要解决关键的两个问题，一个是怎么放，一个是怎么听。

首先，声音怎么播放的问题。在 VR 中制作声音时，要以用户为中心，在整个球形的区域内安排声音位置，确定某一方向基准后，画面内容与用户位置也就是相对确定的，以此来定位，既有水平方向的环绕声，也有垂直方向上的声音。通过水平转动和垂直转动这两个参数，就能控制视角在 360 度球形范围的朝向，以及与画面配合的声音的变化。

另一方面，用户只有一副耳机，要实现电影院里杜比全景声的效果，需要用到一项技术叫做 HRTF (Head-related Transfer Function “头部传送函数”)，该技术能够计算并模拟出声音从某一个方向传来以及移动变化时的效果，类似于一个滤波器，对原始声音进行频段上的调整，使其接近人耳接收到的听感效果，并通过耳机来回放。

基于这样的原理，不少厂商已经进行了尝试来创造 VR 中的音效。

Oculus

早在 2014 年，Oculus 授权 VisiSonic 的 RealSpace 3D 音频技术，并将其融入 Oculus Audio SDK 中。通过跟踪器上所发来的空间信息来处理声音信息，让听者觉得该声音是从这个物体中发出来的。这项技术非常依赖定制的 HRTF，通过耳机来再现精准的空间定位。

NVIDIA

到 2016 年 5 月，NVIDIA 就推出了一个专门用于虚拟现实场景，第一个基于物理技术的声学仿真技术“VRWorks Audio”，借鉴了光线追踪渲染的思路，充分考虑了 3D 场景的渲染，通过将音频交互映射到 3D 场景中的物体上，使音频听起来更加自然。用户不断移动，能够

听到回声的变化以及带来的空间感，除了能够判断声音是由该物体发出之外，还能判断出物体的方向、远近等等跟多的信息。

AMD

与英伟达类似，2016年8月，AMD在发布了一项名为TrueAudio Next的实时动态声音渲染技术，让虚拟现实中的声音和画面更为同步。

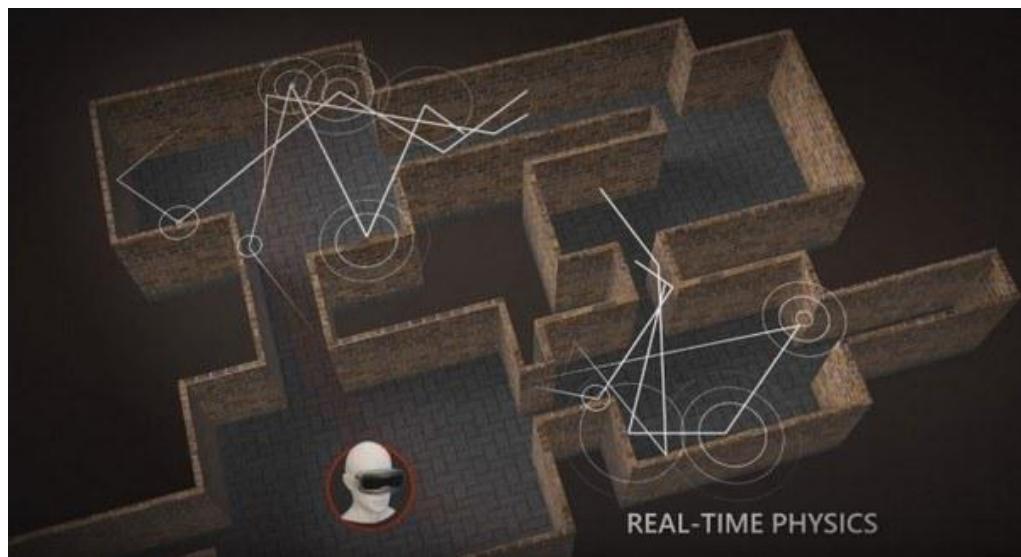


图 2.56 实时动态声音渲染技术

该技术同样使用物理方式模拟，让渲染的声音无限接近真实环境的声音，在虚拟建模中进行多次反射，利用Radeon Rays光线追踪技术让系统辨别VR空间布局并定位空间中的物体。AMD已将该技术开源。

谷歌

近年谷歌也与音频公司Firelight和audiokinetic合作，推出一个VR音频插件。开发者利用该插件可以根据虚拟空间大小、材料以及对象位置的改变来调整声音，营造更加逼真的氛围。该插件可以无缝集成到Unity和Unreal引擎中，使用时开发者只需要对3D音频进行简单调节，能够很轻易地创造空间音频。

此外，谷歌公布了面向Web端的Omnitone，一个跨浏览器支持的开源空间音频渲染器。同样使用HRTF，但是他们主要解决的问题是，在已有的浏览器里引进环绕立体声技术，同时不能干扰浏览器原本的运行。

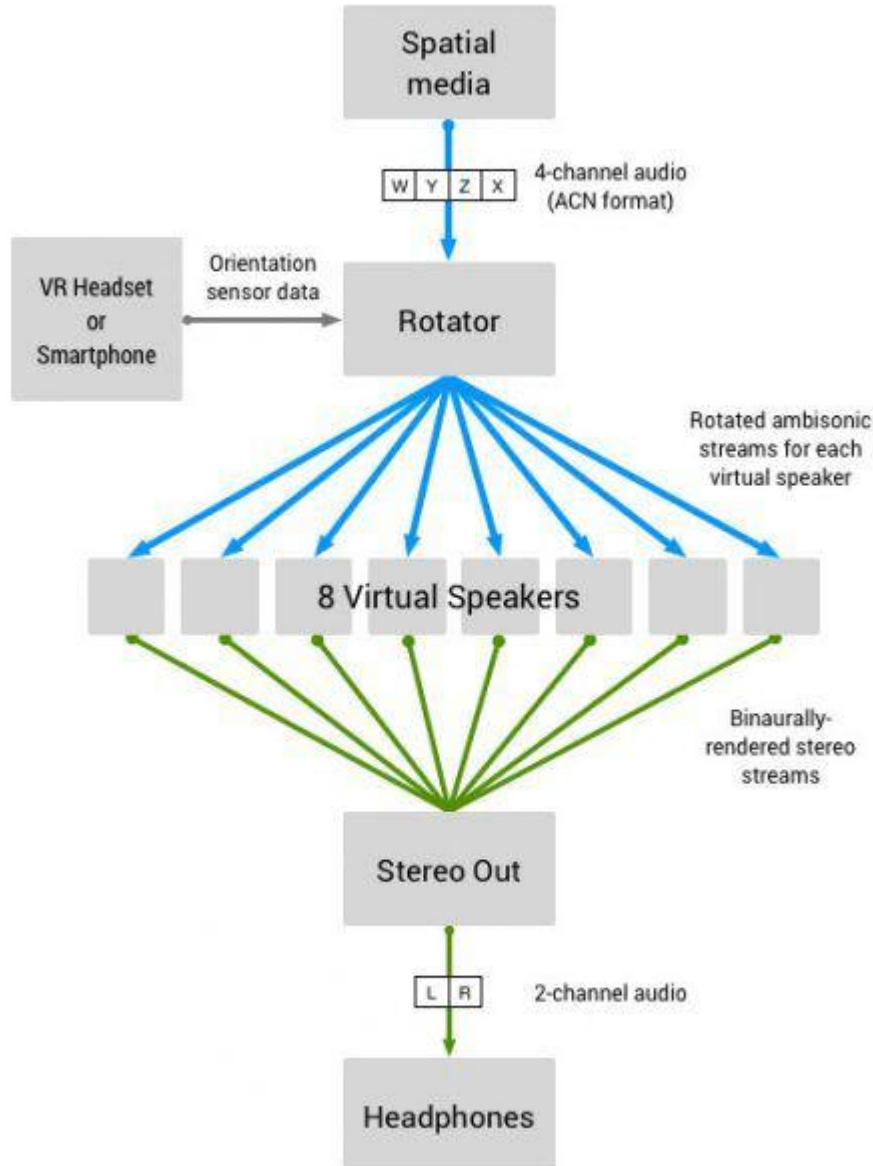


图 2.57 谷歌 Omnitone

上图是谷歌 Omnitone 的解决方案。在环绕立体声里包含了 4 种声道，可在任一扬声器中解码。谷歌在 Omnitone 中设置了 8 个虚拟扬声器来渲染双耳音频流，将 VR 头显中的方向传感器数据与解码器无缝衔接，完成音场转换，从而让用户通过耳机就能体验到空间感。

Valve

Valve 此前曾收购了音效公司 Impulsonic，Impulsonic 有一个基于物理的声音传播和 3D 音频解决方案，名为“Phonon”，近日，Valve 开放了 Photon 音效工具的后续产物 Steam Audio SDK。该方案能够通过空间音效增强 VR 沉浸体验，允许游戏的音频与场景几何体建立交互与反弹回音，从而增强体验。Steam Audio 支持 Windows、Linux、macOS 和安卓等多个平台，也不局限于特定的 VR 设备和 Steam。

小结

目前已有的音频技术可以实现 360° 全景声，可以通过声音辨别方向、距离。但是 VR 音频技术要求不仅仅能够在提供 VR 环境中物体的位置信息，更要反馈出更多的空间环境状

态。

以 VR 游戏为例，当光线越来越暗，视觉必定受到限制，这个时候就要靠音频来确定环境状态，脚步声、风声、动物的叫声等都能为玩家提供信息，诱导下一步的行动和交互。因此，精准有效的音频技术在 VR 中特别重要，不仅仅是游戏、视频，还有其他例如教育、社交等领域，VR 音频技术也需要进一步的成熟。

2.6 数据转换技术

2.6.1 SLAM

即时定位与地图构建 (Simultaneous Localization and Mapping, SLAM) 是根据多点云构建 3D 地图 (点云) 的过程，每个点都由不同的定位 (扫描仪位置) 获取，但不必精确地知道这些定位。SLAM 的作用是融合这些点云，逐渐更新预设的定位，直到不同点云之间的距离最小化。图 2.58 (左) 显示了同一场景中两个未对准的点云，图 2.58 (右) 显示了 SLAM 对齐的效果，这有效地增加了所得点云的密度。

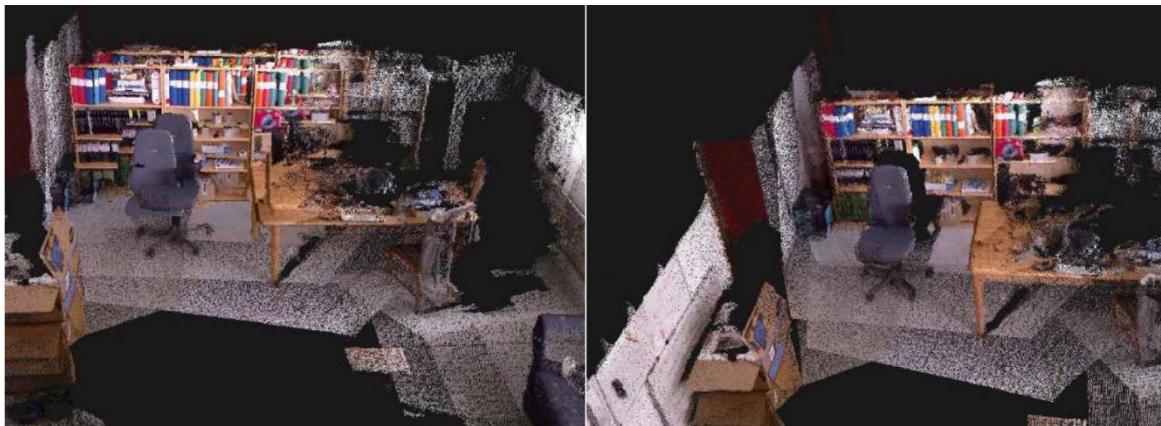


图 2.58 对齐前 (左) 和对齐后 (右) 的点云的 SLAM 融合

SLAM 在激光雷达扫描中非常有用，其中点云的密度随着扫描物体与扫描仪定位之间的距离而减小。为了获得更均匀的点云，可以从不同位置扫描场景，并使用 SLAM 将获取的点云融合在一起。然后可以对得到的高密度点云进行编码等操作。

2.6.2 点云到深度图

如图 2.59 中所示 (右上)，由激光雷达扫描仪捕获的点实际上是以球坐标 (即角度和深度) 获取的。大多数情况下，这些坐标在扫描仪内部转换为笛卡尔坐标 (x, y, z)，参见图 2.59 (左下)。

然而，笛卡尔坐标仅是中间坐标 (可能在 SLAM 预处理之后)，最终仍会被转换回投影深度图，表示每个点到平面的距离。图 2.59 (右下) 显示了这种深度图，以及进一步用于编码的低通和高通二次采样版本 (小波分解)。

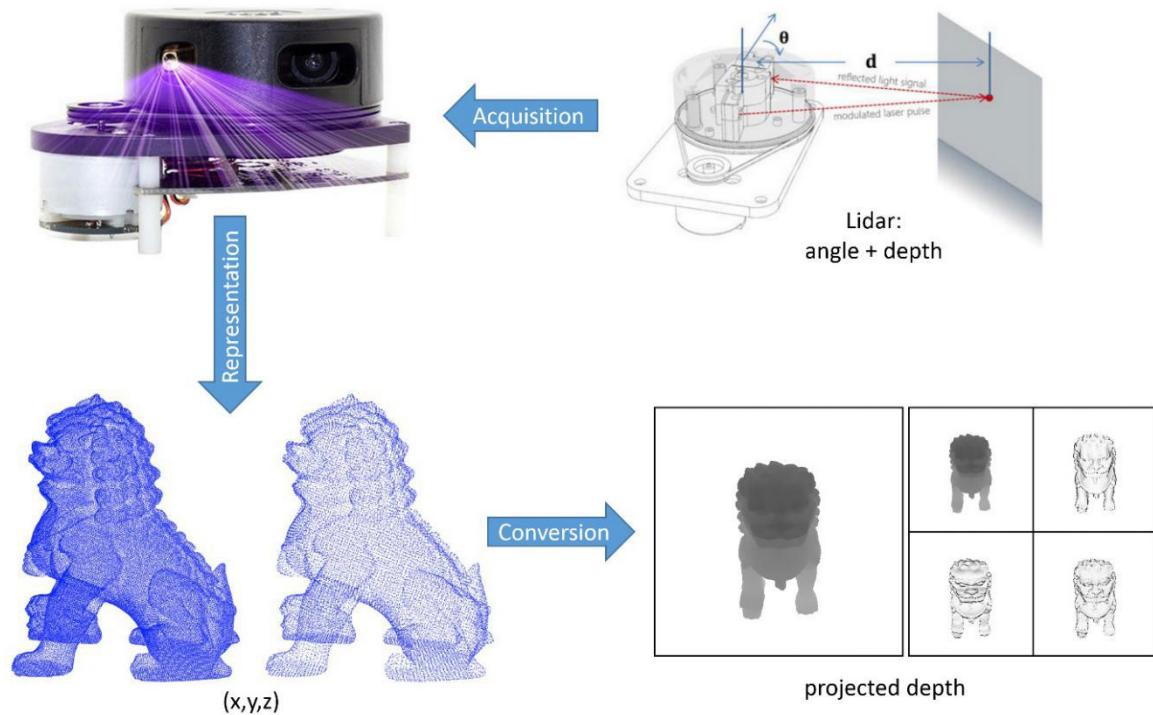


图 2.59 激光雷达获取（角度+深度）点云通常用（ x , y , z ）表示，但也可能转换为投影深度图。

2.6.3 点云到三角形网格

图 2.60 显示了将点云转换为三角网格的效果，这样便可以在经典的 OpenGL 处理通道中轻松渲染。要注意的是，良好重建封闭区域是较为困难的，这一点可以通过上述的 SLAM 技术来处理。

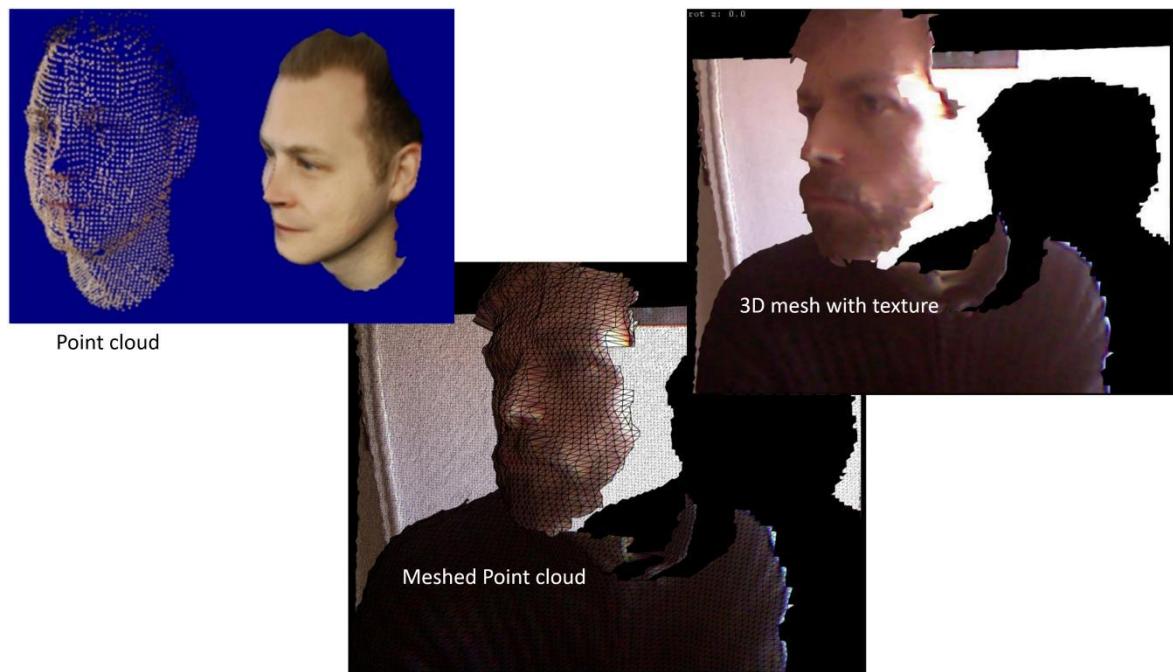


图 2.60 将采集的点云转换为 3D 网格

2.6.4 点云到平面基元

在某些应用中，例如建筑信息模型（BIM），提取表示对象的平面基元比保持点云的各个点更有益，如图 2.61 所示。进一步而言，将多个点归并至同一个平面基元的做法也可以在基于网格的编码中产生更好的性能。

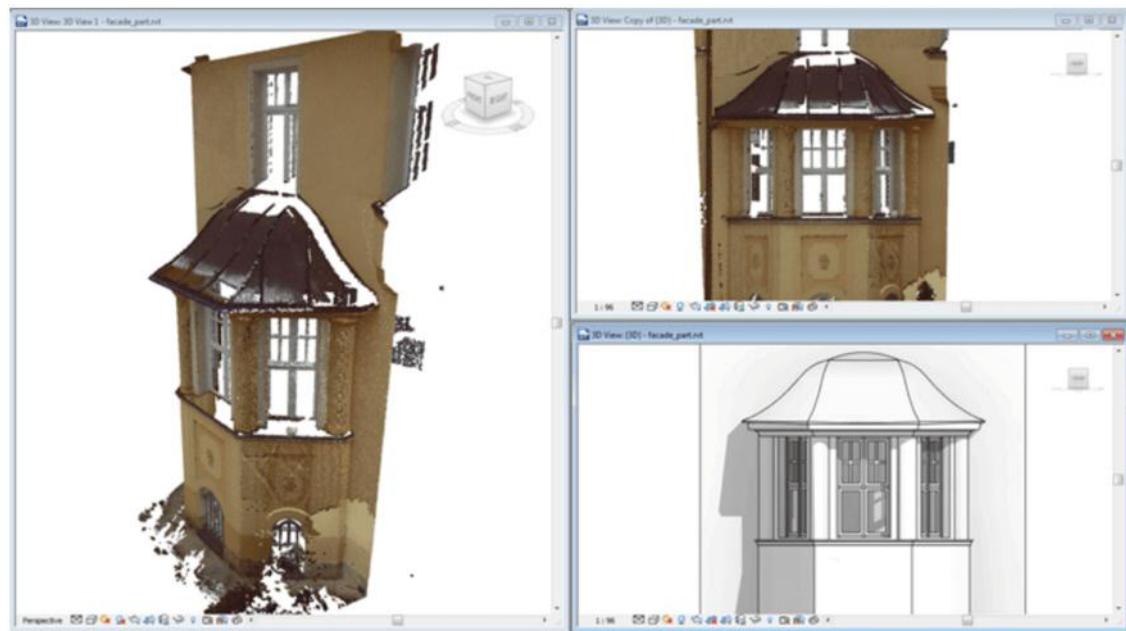


图 2.61 点云（左）的平面基元提取（右下）

为了编码上的目的，图 2.62 的背景与前景对象被分离开，并将背景拼接成一个可用作全景图像的纹理图，这在 VR 中经常进行，只有这样大尺寸图像的窗口区域才能在 HMD 设备上可视化，参见图 2.63。图 2.62 的其余点则可以被视为常规点云进行后续操作。



图 2.62 根据深度信息图 (a) 和投影视图 (b) 的混合点云和分段平面表示 (c)



图 2.63 虚拟现实的全景纹理图：在任何时间戳，纹理图的一个区域在头戴式设备中呈现

2.6.5 点云与光场

通常，点云中的点不需要在所有方向上具有均匀的颜色。实际上，在自然界中，许多现象是镜面的，在所有方向上都具有不均匀的颜色分布。这样的信息可以通过所谓的 BRDF 函数来表示，以告知点云中的每个点在每个方向上的颜色传播。从某种意义上说，这个点本身就会发出一个“光场”，所有点拥有的场的结合会产生一个所谓的光场。这清楚地表明了点云和光场之间的双射等价关系。

如图 2.64 所示，点云/光场表示空间中的每个立体像素 (x, y, z) ，其颜色 C 在各种光线方向 (θ, ϕ) 下会有所变化，即颜色 C 是 5 参数的函数：

$$C = f(x, y, z, \theta, \phi) \quad (2.8)$$

然而在实际中，该函数可以减少至 4 个参数，因为光线一般在整个传播过程中保持不变（除非存在微粒，如雾，这会降低传播路径上的光强度）。因此，每条光线可以通过其与相机平行的平面的交点 (s, t) 及其传播角度 (θ, ϕ) 来表示：

$$C = f(s, t, \theta, \phi) \quad (2.9)$$

光线也可以用两个平行平面的交点 (s, t) 和 (u, v) 表示，见图 2.64：

$$C = h(s, t, u, v) \quad (2.10)$$

后者是文献中最常使用的光场数据表示法。

总之，除了利用点 (x, y, z) 和方向 (θ, ϕ) 表示对应的颜色外，也可以用它与两个平行平面的交点 (s, t) 和 (u, v) 表示从这一点发出的光线。

此外，还可以在图 2.64 中观察到点云或光场表示的等效性，以及从摄像机视点的纹理图呈现的深度信息。

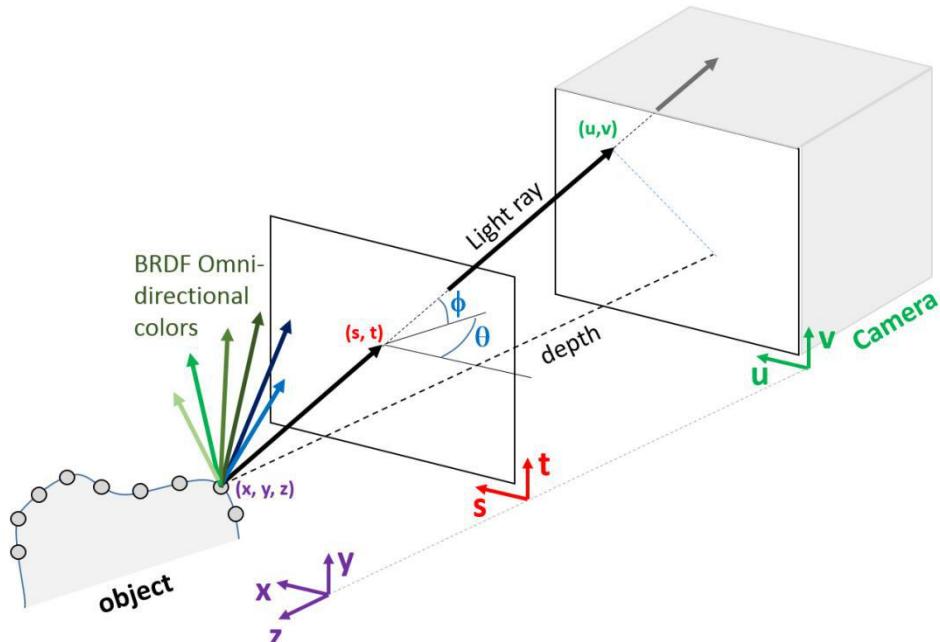


图 2.64 点云中的点发出的光线

2.6.6 光场到深度映射

如前一小节所述，深度信息可以从光场中恢复。这种深度信息通常用于基于深度图像的渲染（DIBR）。

考虑置于拍摄场景前的线性相机阵列的简单情况。将所有位置拍摄的图像堆叠在一起（沿着 u 轴）便创建了图 2.65 上部所示的图像堆栈。

这种图像堆栈的一个水平切片对应于具有许多对角线的所谓的核面图像（EPI），如图 2.65 底部所示。每条线表明了立体元素从一个摄像机镜头跳到下一个摄像机镜头时是如何移动的。远处（大深度）的立体元素几乎不会移动，在 EPI 中形成几乎垂直的线，而前景中的像素在不同角度的镜头中将表现出大的差异，因此产生大斜率的对角线。因此，EPI 中的对角线的斜率提供关于场景中各像素的深度信息，这种方法也被证明对于稀疏相机布置条件下的虚拟视图创建是有用的。

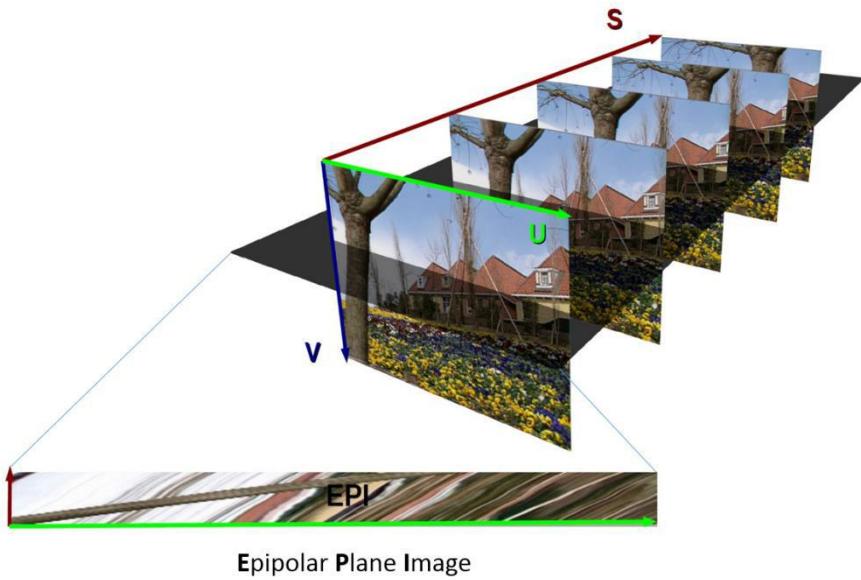


图 2.65 图像堆栈的 EPI 部分

然而，这种方法的一个难点是获得 EPI 的密集表示，如图 2.66（右），当仅有有限数量的摄像机视图，即仅有 EPI 的二次采样版本时，如图 2.66（左），需要利用到特殊的线检测和插值技术。

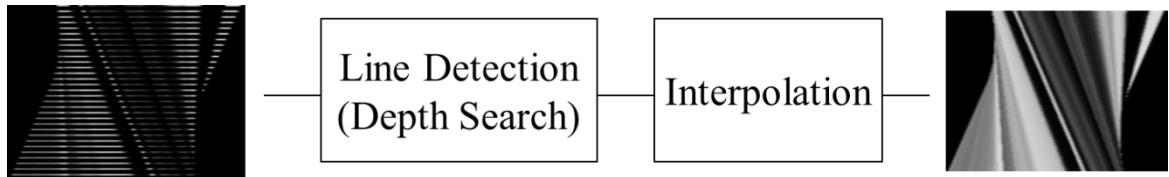


图 2.66 EPI 二次采样版本（左）的直线斜率检测（右）

点云用于沉浸式媒体的优势

- 1) 时效性强：不需要耗费大量人力和时间来制作虚拟的 VR 场景，空间数据采集完毕，稍加处理即可投入使用。
- 2) 真实性强，具备真实空间坐标信息，可对任意位置进行量测；信息采集于真实场地，可谓是对现实空间的完美复制。
- 3) 时空意义，空间信息随着时间的变换而发生改变，点云能够存储不同时期同一空间位置的准确信息。

本章参考资料：

- [1] Jhag, Pirogg (1 July 2011). "Gambling Trends for Online Casino". Retrieved 10 June 2018.
- [2]https://en.wikipedia.org/wiki/Head-mounted_display
- [3]<https://blog.csdn.net/u014640129/article/details/23363661>
- [4]<https://blog.csdn.net/liulong1567/article/details/50457730>
- [5]<https://blog.csdn.net/liulong1567/article/details/50421643>
- [6]<https://github.com/sjtu-medialab/VirtualReality-Chinese/tree/master/VirtualReality>
- [7]<https://zhuanlan.zhihu.com/p/35679418>

- [8]<https://zhuanlan.zhihu.com/p/27551369>
- [9]<https://blog.csdn.net/cbbbc/article/details/70071240>
- [10]<https://blog.csdn.net/nikoong/article/details/79776873>
- [11]<https://blog.csdn.net/shenzi/article/details/5417488>
- [12]<https://blog.csdn.net/dabenxiong666/article/details/55062609>
- [13]<http://smus.com/vr-lens-distortion/>
- [14]http://www.sohu.com/a/41581937_105527
- [15]<http://vrguy.blogspot.com/2016/04/time-warp-explained.html>
- [16]https://blog.csdn.net/fr_han/article/details/50968110
- [17]<http://vga.zol.com.cn/577/5776982.html>
- [18]<https://www.cnblogs.com/Anita9002/p/4975242.html>
- [19]<http://www.tj108.cn/a/2017/1001/22196.html>
- [20]<https://blog.csdn.net/caozhaodan/article/details/77647847>
- [21]<https://docs.unrealengine.com/en-us/Engine/MediaFramework/HowTo/FileMediaSource>
- [22]http://blog.sina.com.cn/s/blog_142f5d2240102xqvu.html
- [23]<http://www.vrzy.com/vr/26418.html>
- [24]<https://mpeg.chiariglione.org/standards/mpeg-i/technical-report-immersive-media>

第三章 全景视频处理技术

3.1 全景图编辑处理软件

在将全景图拍摄制作完成以后，就涉及到对全景图的编辑处理，常用的全景图编辑软件有 Adobe 公司的 Photoshop 和一些小的全景图编辑平台。

Photoshop 是一款设计制作和编辑图像的实用工具，它的功能十分强大，是集图像扫描、编辑修改、图像制作、广告创意等功能于一体的图形图像处理软件，旧版本的 Photoshop 中（PS 2015.5 以前的版本），用户很难通过该软件编辑制作全景图，而且即使做出来，效果也不尽如人意。在 PS 的新版本中，添加了全景图处理的模块，可以在该软件中制作编辑全景图。



图 3.1 PS 开始界面

可以在初始界面直接将制作好的全景图加载进来，如下图所示：



图 3.2 加载全景图

接下来就可以创建全景操作空间：3D-球面全景-导入全景图，然后会弹出全景空间的设置项，根据需要设置相应的宽度和高度即可，完成后点击确定。



图 3.3 创建全景操作空间

随后就可以看到一个 3D 图层，并且默认是可以 360 度拖动的。这时，全景图片的处理就变得和平面图片处理一样轻松了。

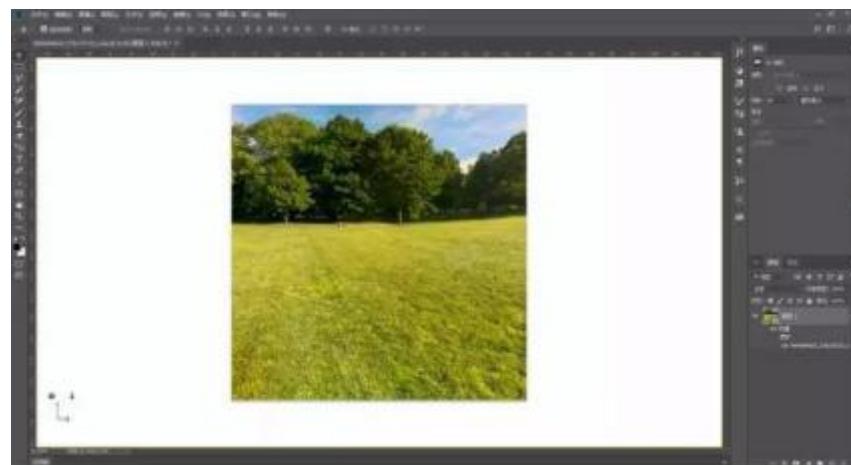


图 3.4 处理全景图片

接下来就可以进行像平面图一样的编辑操作，可以将底部或顶部的盲区调整至居中位置，然后利用仿制图章工具来修补盲区；也可以修改拼接错缝的区域，具体操作就是首先用多变形套索工具复制一层， $Ctrl+T$ 选中复制图层，应用变形操作，用鼠标调整位置将其对齐，调整完毕后选中图层并向右合并即可；可以在全景图中添加 3D 物体，填充所需的物件，达到所想实现的效果，具体操作就不再过多介绍；可以添加文字，将文字做成 3D 效果。



图 3.5 添加文字效果

全景图的编辑主要就是通过 PS 软件操作，下面是有关全景视频的编辑介绍。

3.2 全景非线性编辑软件

Premiere

首先介绍一下非线性编辑软件，它通常是指用电子手段按要求先用组合编辑将拍摄的素材按顺序编成新的连续画面，然后用插入编辑对某一段进行同样长度的替换，但是想要去除、编辑加长中间的某一段就不可能了，除非将那一段以后的画面全部重录。非线性的主要目标是提供对原素材任意部分的随机存取、修改和处理。它的真正推动力来自视频码率压缩。码率压缩技术的飞速发展使低码率下的图像质量有了很大的提高，推动了非线性编辑在专业视频领域中的应用。

前文在将 PS 的全景图编辑功能介绍以后，下面介绍 Adobe 的另一款软件 Premiere，该软件的主要用途是视频的剪辑、编码、转换格式和制作一些简单的效果，而且在较新版本中加入了 VR 的预览功能和一些 VR 效果。

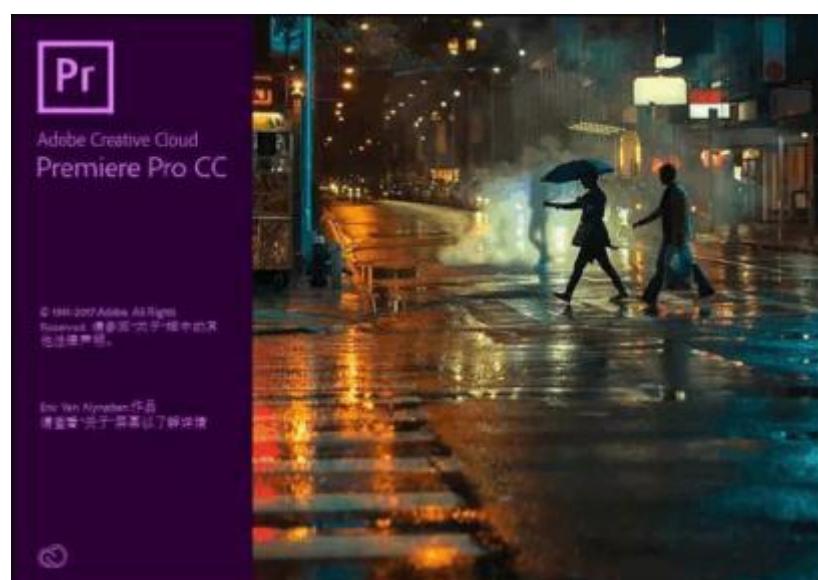


图 3.6 Pr 开始界面

首先可以将一段全景视频导入到 PR 中，在预览窗口右键找到 VR 视频并点击“启用”即可进入可拖动的全景模式，并且在窗口的下方和右侧分别由两个控制轮以及滑块显示控制，当电脑连接头显时，可以在 VR 头显中进行预览。



图 3.7 导入全景视频

在 PR 中，VR 编辑功能全部放在了视频效果里面，基本功能也和后面要介绍的 AE 大致相同：



图 3.8 VR 编辑功能

当为视频增加字幕的时候，需要将视频效果中沉浸式视频的“VR 平面到球面”选项拖至字幕图层，这样在全景模式下文字就不会出现畸变效应了，然后通过调整不同的参数就能够得到不同的效果，例如可以将缩放调大（度数越大字幕越近），而且结合关键帧还可以做很多的动态字幕效果；同样的方法还能在视频中进行贴图，贴视频等操作，不过需要了解相关设置的含义，比如羽化边缘，调整旋转源以及旋转投影的数值等，关键是调整数值使得贴图与全景视频中的墙面平行贴合。

After Effect

接下来再介绍一下另一款更加强大的软件 After Effect，该软件主要是用来影视后期制作的后成，甚至很多大制作的电影都是用 AE 软件进行后期制作的，可以实现很多特效。



图 3.9 Ae 开始界面

在使用该软件时，直接导入想要编辑的全景视频，同样的类似 PR 在效果-沉浸式视频的选项卡中一共有 12 项 VR 编辑功能，可以将这些功能大致分为三组：

转换：VR 球面到平面，VR 平面到球面，VR 旋转球面，VR 转换器

降噪 / 锐化：VR 降噪、VR 锐化、VR 模糊

画面处理：VR 数字故障、VR 分形分色、VR 颜色渐变、VR 色差、VR 发光

下面将逐一介绍这些功能。

VR 球面到平面

这个功能就是将有球形畸变的视频内容转换成平面内容，下面是它的操作面板：



图 3.10 VR 球面到平面

预设的输入格式有三种：球形图，立方图和球面投影。输出帧宽度：有 256~16384 的可调范围，通常不超过原始素材的宽度，盲目增大并不会变清晰，和插值像素的道理一样。输出帧速率应该是帧的比率，配合上面的输出帧宽度完全就可以获得想要的平面尺寸。旋转投影是改变三个轴的旋转方向，改变以后主视角随之更改。反旋转就是将度数取反。

另外还可以将关键帧做成环视效果，在起始位置和结束位置分别创建关键帧，然后播放即可。

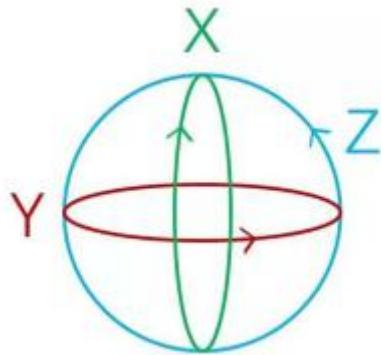


图 3.11 XYZ 轴旋转方向

VR 平面到球面

该功能使 2D 图片或者视频应用于全景空间中但不变形，下面是它的控制面板：



图 3.12 VR 平面到球面

预设有两种布局：单像和立体，普通视频选单像，3D 选立体，缩放是在全景模式下的远近，不是大小， 180° 时与观看的人位置重合， 0° 时无穷远。立体视差只有选择 3D 模式下才有效，羽化即是边缘透明度，旋转源可以理解为自转，旋转投影为公转。

VR 旋转球面

对应在全景空间中的 XYZ 轴向旋转，可以调整全景视频的水平等方向至合适的方向，以满足特定的观赏体验，下面是操作面板：



图 3.13 VR 旋转球面

VR 转换器

各种类型的的媒体格式的输入输出转换，下面是它的操作面板：



图 3.14 VR 转换器

大体的操作是一样的，关键是输入和输出转换，一共有 9 种输入类型和 8 种输出类型：



图 3.15 VR 转换器的输入和输出格式

不同的输入输出组合能够产生不同的效果，可以根据自己的需求进行选择即可。

VR 降噪

降噪就是细节模糊，当光线较弱时，特别是夜景拍摄时很容易在图像中产生噪点。VR降噪大多应用于弱光环境，也可以用来调制清新柔和的视觉风格，默认的杂色级别是0.2，可以根据实际画面自行调节。



图 3.16 VR 降噪

VR 锐化

锐化就是消除误差大的像素，能够适当的提高画面清晰度，但过度提高会造成画面失真，下面是控制面板：



图 3.17 VR 锐化

VR 模糊

模糊是一般用来做开头的片场：可以由模糊到逐渐清晰，也可以进行其他艺术创作比如通过局部模糊的方式引导观众的关注点，也可以通过调整关键帧变换视角。



图 3.18 VR 模糊

VR 数字故障

该功能是一种类似于电视信号故障的艺术表现形式，但是一般用不到这个功能。应用后的效果如下图：



图 3.19 VR 数字故障

VR 分型杂色

下面是其功能面板：

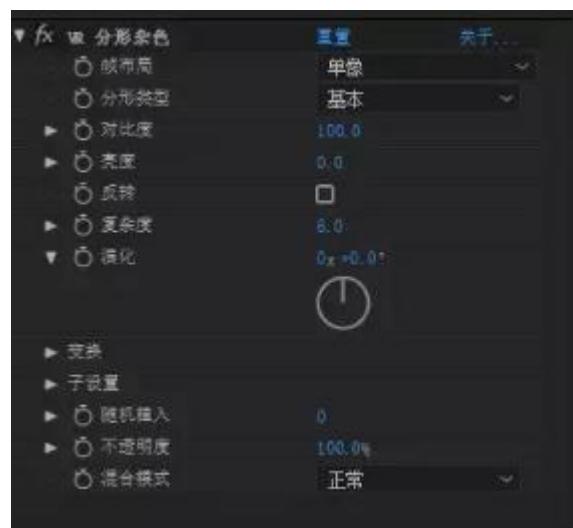


图 3.20 VR 分型杂色

分型类型总共有 4 种，具体图示如下：

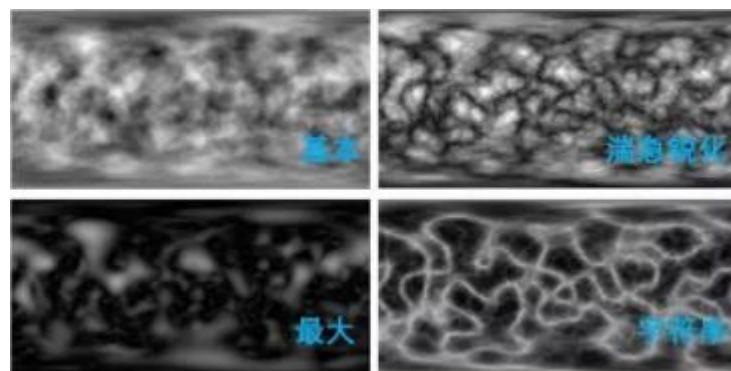


图 3.21 VR 分型类型

对比度：图中黑白反差，对比度越大，线条越清晰；

亮度：图中黑白占比，亮度越大，白色越多；

复杂度：细节越少，数值越低，分型图像越模糊；

演化：分型演化过程；
 变换：同 VR 旋转球面；
 子设置：分型自身的 XYZ 旋转；
 混合模式：两个图层的混合效果；
 根据这些设置可以做出云雾等不同的变换效果。

VR 色差

下图是该功能面板，当色差相等时，画面就会呈现实色，并且有一个远近关系的变化，可以用来做转场效果等。



图 3.22 VR 色差

VR 颜色渐变

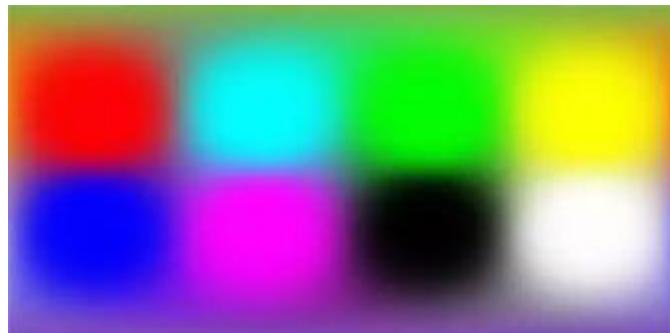


图 3.23 VR 颜色渐变中色块

点数：即上图中色块的个数；
 渐变功率：每个色块的羽化值大小；
 渐变混合：相邻颜色间的过渡范围大小；
 点：上图中每个色块的颜色；
 混合模式：同 PS；
 下面是控制面板：



图 3.24 VR 颜色渐变

VR 发光

发光就是使画面中近乎白色的部位呈现光扩散，可以通过调节亮度阈值来达到转场效果，下图是控制面板：



图 3.25 VR 发光

DaVinci Resolve

上面介绍了 Adobe 公司关于全景视频的编辑以及相关软件，编辑功能主要是对原来的视频做出一些制作者想要实现的效果。常见的非线编软件还有 DaVinci Resolve，其有强大的调色系统集与编辑功能，能够帮助用户进行视频剪辑、调色、精编和交付等处理。该软件拥有非常多的实用工具，制作速度快，稳定的兼容性以及一流的画面质量，并且是许多好莱坞电影的首选解决方案。

该软件是一个免费软件，关键是在编辑全景视频的时候确保视频的分辨率和导出时的分辨率是相匹配的，如图 3.26：

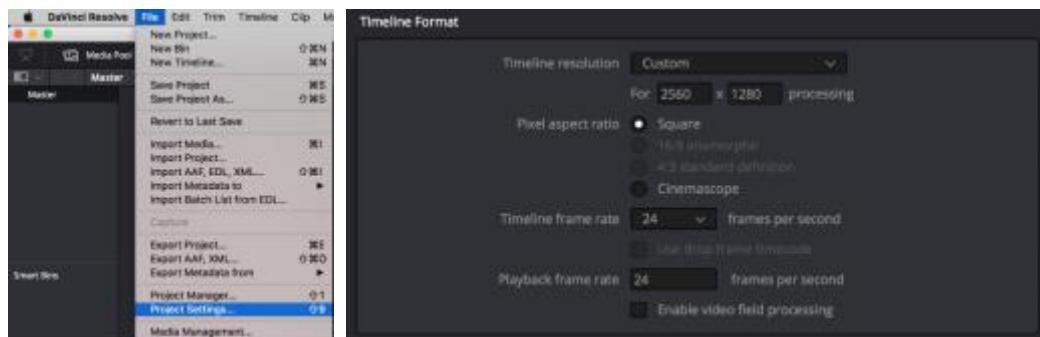


图 3.26 DaVinci 基本设置

初始设置好之后即可导入 equirectangular 视频文件，将文件添加到时间线，然后就可以像编辑其他任何视频一样进行编辑，添加制作者想要达到的效果，还可以设置视频格式和编解码器，最后渲染导出即可。

3.3 视频稳像技术

在之前章节提到过拍摄全景视频的相机或视觉系统，以及全景视频的成像原理。至于全景视频的拍摄方式，对于专业级的设备，毫无疑问都是采用固定拍摄的方式，因为在图像拼接的过程中，每个镜头获取的光场信息，图像的亮度、色调等等一定要保持一致，否则在之后的特征点匹配、视差处理、图像融合的过程中，误差逐渐放大，在生成的图片及视频中会出现明显的接缝甚至畸变。对于体验级的设备，很难去要求用户采用固定拍摄的方式，而且大多数用户也不满足于固定拍摄的效果，往往会选择手持式移动拍摄的方法。摄影者的移动以及手的抖动会引起相机的摆动，或多或少会产生上述的一些问题，除此之外，画面的抖动会非常影响视觉舒适度和眩晕感。而且，全景相机很少会像普通的相机一样加入防抖技术，

再加上在全景视频中用户可以全方位地观看视频内容，对抖动的敏感度要远高于普通的视频，所以，针对于运动条件下拍摄的全景视频，进行视频稳像的处理来降低拍摄画面的抖动是很有必要的。

视频稳像简介

视频稳像，顾名思义，就是对拍摄的视频进行稳像处理，使得原始视频中抖动的画面变得平稳，而尽量不损失画面的清晰度。稳像技术基本分为三类：机械稳像、光学稳像、电子稳像。机械稳像的主要原理是通过一些传感器如陀螺仪来获取相机的运动，相机的处理器控制图像传感器按相反的方向移动，对相机的运动做补偿；光学稳像依靠特殊的元件根据镜头的抖动方向和位移量加以补偿，以得到稳定的图像。这两种稳像方法的缺点在于代价较大、设备携带不便，且稳像效果不够好，难以满足如今视频稳像的需求。而电子稳像通过估计相机的运动路线，在计算机或者其他设备上将视频中的每一帧画面进行移动，使得输出的视频中的运动是平滑的。电子稳像具有更容易实现，更精确，更灵活，成本更低等优点，是目前视频稳像领域的主流研究方向。

目前大部分的电子稳像都属于基于运动的算法，适用于抖动较为平缓的场景，比如平移运动、小角度的旋转运动等；亦或是视频中运动物体目标较小，容易跟踪。算法主要分为三个步骤：全局运动估计、运动补偿和图像生成，如下图所示。全局运动指的是处于主导地位的像素运动，也可以表示为相机的运动，根据如何估计全局运动可以将算法分为 2D 稳像和 3D 稳像两类。运动补偿是算法的核心，指的是从较为抖动的全局运动中分离出抖动和主观运动，并从中分离出主导运动。图像生成指的是将原始视频做处理，输出稳定后视频的过程。

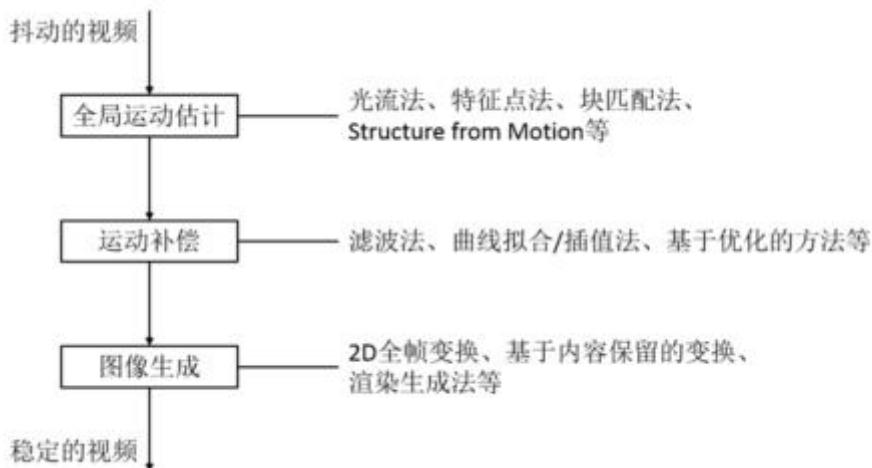


图 3.27 基于运动的电子稳像算法的基本处理流程图

2D 稳像算法通过相邻帧之间的运动模型来表示相机在二维平面内的全局运动，常用的方法是特征点法。这些方法仅考虑二维平面内的相机运动轨迹，与实际相机的运动可能会有些许偏差；3D 稳像算法主要依靠 Structure-from-Motion (SfM) 进行，通过 2D 图像序列来计算图像中物体的 3D 结构的技术，这种算法复杂度非常大，且适用场景较少，比如视频中缺少视差信息，拍摄中镜头有缩放等等情况，很难计算出有效的三维结构。

稳像算法

以前述基本原理为基础，研究人员不断提出性能更优的视频稳像算法。代表性的算法之一是 Grundmann 等人 2011 年提出的基于 L1 范数的视频稳像算法，该算法性能突出，复杂度不高，后来被集成应用于 Youtube 上载视频的在线编辑软件中，成为当前视频稳像算法研究的 benchmark。性能继续的提升方向有多种途径：一个思路是设计更合理的重建路径和优

化目标，如用 L1+L2 范数代替 L1，可以保留更多的场景内容，减少黑边；另外一个思路是利用与视频同步的位置/姿态传感器数据，直接估计出运动参数，进而降低 SfM 的难度，该类技术特别适用于智能手机上的视频稳像。

然而，一般的视频稳像算法并不适用于全景视频，因为全景视频覆盖水平垂直 360° 的内容，然后用 Equirectangular 投影方式（ERP）将三维球形画面映射到二维平面上。所以说 ERP 投影得到的视频中的抖动并不能表示相机的运动，无法得到准确的全局估计。因此，针对全景视频的稳像技术研究是一项重要且具备挑战性的研究。

针对移动多相机平台的全景视频稳像算法

实际上，多相机系统在拍摄全景视频时，不仅仅产生全局运动，同时还会因各相机间的差异以及视差引起的深度错觉而产生较为独立的抖动。针对这一特点，Ameer Hamza 等人提出了一种针对移动多相机平台的全景视频稳像算法。该算法将拼接的视频帧和相应的混合掩模（Blend mask）作为输入，并最小化由所有三种类型抖动所引起的视频质量下降的程度。

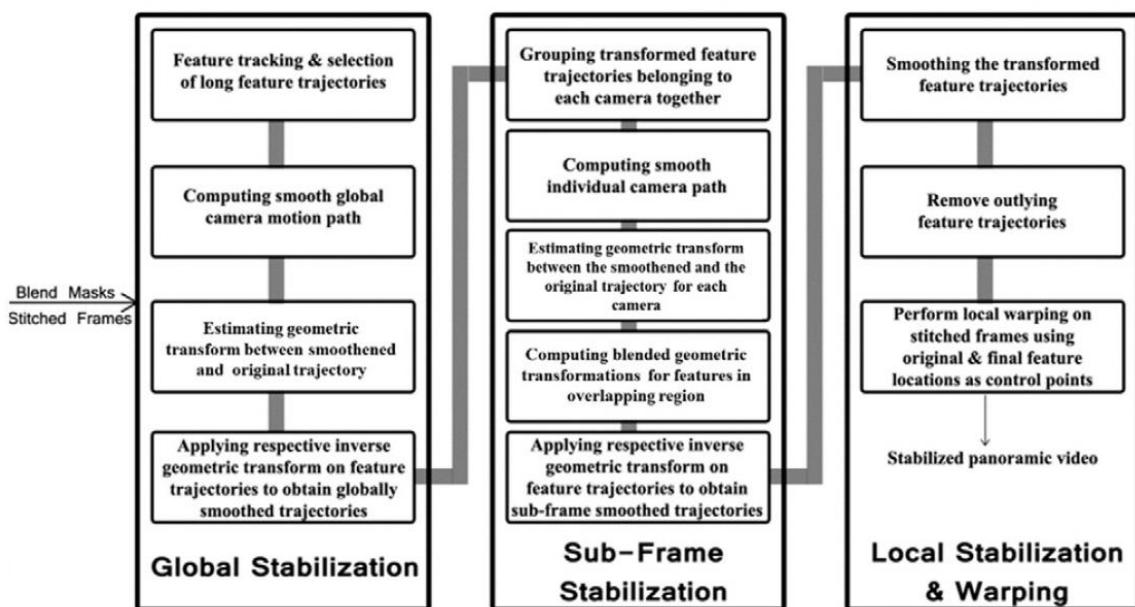


图 3.28 全景视频稳像算法流程图

图 3.28 提供了该算法的流程图。假设对一个由 C 个摄像机拍摄的 K 帧全景视频进行处理，算法则需要全景帧 ($P_0, P_1 \dots P_K$) 和 C 个混合掩模 (Bm^1, Bm^2, \dots, Bm^C) 作为输入。我们首先从全景视频帧中提取显著特征轨迹，并使用它们来估计多个步骤的各类变换，以最终呈现出平滑运动的视频。

首先，从完整全景区域提取的特征轨迹是平滑的，并用于估计全景视频帧的全局几何变换。虽然在这个阶段会对每个像素应用计算的变换，但是这些几何变换仅是用于处理全局抖动的，仍需要进一步的变换来处理其他两种类型的抖动。由于后续步骤需要对特征轨迹进行处理，因此中间过程的变换均会被集中到最后一步进行，以避免在每个中间稳定步骤之后重新计算轨迹。因此，每个全景帧的全局变换仅应用于特征轨迹而不是应用于所有像素。

在第二个步骤中，这些变换的轨迹根据相应的摄像机运动进行分组，并且用于估计各个摄像机捕捉区域的几何变换。然后根据相应估计的变换来转换每个相机组的轨迹。要注意的是，每块全景重叠区域中的轨迹具有两个估计的变换，分别用于两个相邻的相机。因此，将混合掩模加权变换应用于这些轨迹可以确保两个相机空间的平滑过渡。

此时，变换轨迹中剩余的主要残余抖动源是由视差引起的抖动。为了处理这一抖动，每

个轨迹独立地进行平滑，接着是对各轨迹进行聚类，形成经历类似抖动并因此属于相同深度平面或具有相同移动属性的特征组。不符合任何这些组的特征轨迹被标记为错误轨迹。最终使用非错误特征轨迹的原始位置和变换位置来扭曲原始帧，即有效地共同应用所有三个稳定变换以产生稳定的全景视频。

Facebook 3D-2D 算法

针对全景视频的稳像，Facebook Research 的 Johannes Koef 提出了一种混合的 3D-2D 算法：利用 3D 分析估计出适当间隔的关键帧的相对旋转；然后恢复关键帧之间的相对旋转，用内插计算对内部帧进行调整；最后对内部帧的旋转进行 2D 优化，最大化特征点轨迹的平滑度。这种算法针对全景视频，结合了 2D 稳像算法和 3D 稳像算法的优点，有着不错的效果。并且，相较于其他的稳像算法，这种算法有着如下的优点：

- (1) 准确性：用 3D 分析的方法估计关键帧之间的相对旋转，不会混淆旋转/平移运动和非静态异常特征等。
- (2) 鲁棒性：关键帧被间隔开，使得帧与帧之间有着足够的运动，可以让 3D 估计有着更可靠的结果。
- (3) 正则化：关键帧之间的真实相对旋转为内部帧的 2D 优化提供正则化基础。这限制了的变形旋转运动模型，防止产生新的摆动伪影，并且对收敛具有强的积极效果。
- (4) 速度：虽然算法对内部帧使用非线性优化方法，但问题具有良性误差函数，良好初始化，并快速收敛。

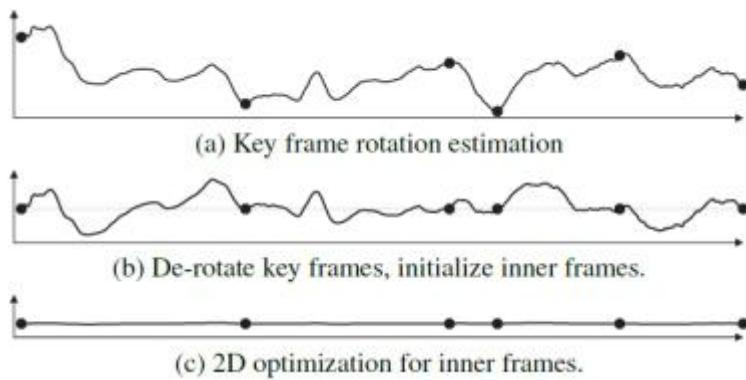


图 3.29 混合 3D-2D 算法示意图

与其他稳像算法类似，混合的 3D-2D 算法第一步是从跟踪特征点的运动开始。因为全景视频的输入多为 Equirectangular 投影方式 (ERP)，这种投影方式在南北两极的地方会产生无限的尺寸拉伸，所以需要将视频帧转换为较少失真的 CubeMap 投影方式（立方图投影），进行特征点的提取和跟踪。整个过程使用 256x256 像素的立方体面，独立于输入分辨率，并且只使用亮度平面进行跟踪。算法使用 Kanade - Lucas - Tomasi feature tracker (KLT tracker) 作为特征点的提取程序，并使用金字塔形 Lukas-Kanade 跟踪算法进行特征点的跟踪。

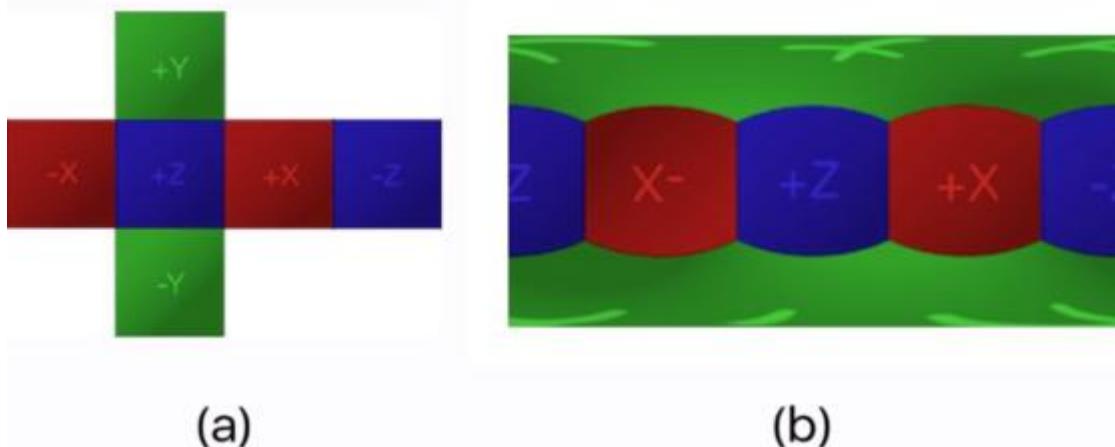


图 3.30 ERP 投影和 CubeMap 投影的对应位置关系

算法当中有一个“关键帧”的概念，它们发挥非常重要的作用，因为整个算法的基础就是估计它们之间的相对旋转，然后形成用于后续 2D 优化的正则化基础。至于如何选取关键帧，算法使用 Shi-Tomasi 算法生成特征点列表，通过递减特征强度排序，排查列表，当一个特征远离任何先前选择或主动跟踪的特征超过 2° 时，将其设为关键帧，并产生一个新的轨道，用于后续的跟踪。

下一步是估计连续关键帧之间的相对旋转。这里利用 OpenGV 库实现了 Nister 和 Kneip 的五点算法：给定两台摄像机拍摄画面对应匹配的五个点对，可以估计出摄像机之间的相对旋转和平移。使用这种 3D 分析的方法，可以从平移运动中区分出真实的旋转，使得估计的结果更加接近实际的运动。

现在关键帧之间的旋转得到了补偿，接下来固定它们之间的旋转，研究内部帧之间的旋转。之前有提到过，这里不再采用 3D 分析的方法，而是使用 2D 优化的方法来稳定内部帧之间的旋转。优化的目标是对于非关键帧，找到最理想的旋转，最大化特征点轨迹的平滑度。在优化过程中，保持关键帧的旋转固定，它们为正则化提供了基础并增加了收敛性。算法将整个优化过程抽象化为非线性最小二乘问题，然后使用 Ceres 库解决它。

完成上述的优化过程，可以消除大部分的相机抖动，但是由于一些小的平移运动、视差、镜头校准、拼接伪影、卷帘快门摆动等，处理后的结果往往还会有一部分残余的抖动。此时需要为运动模型添加一些灵活性来解决这个问题，使得它可以适应并消除轻微的图像变形。同时，模型也不能变得过于灵活，所以需要适当地被约束。该算法设计了一个“变形-旋转”模型来处理上面提到的一些问题。在这个模型中，在单位球体上均匀分布 6 个顶点，分别相差 90° 并将整个球体分割成 8 个全等的球面三角形，每个八分圆一个。对于每个顶点，记录下它的旋转值，然后利用球面重心坐标插值计算这些顶点的旋转，并将其集成到原始的旋转中。最后，将新的运动模型置于稳定优化问题当中，得到最终的稳像结果。

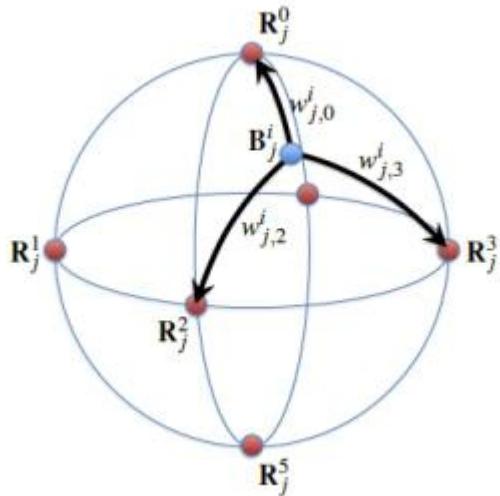


图 3.31 “变形-旋转”运动模型

稳像算法除了处理效果，运行时间也是研究的关键。该算法不像大多数稳像算法一样采用 GPU 加速来减少处理时间，而是仅通过 CPU 来进行处理。虽然 CPU 在计算变换坐标时比较慢，但是整个变换过程是平稳的，为了得到更好的效果，代码实现的过程中作者并未采用 GPU 加速，而是通过计算每个 8x8 像素的变换坐标，然后对其进行双线性插值的方法优化了计算变换坐标的过程。实验采用的 CPU 为 Intel Core i7-5930K CPU，主频 3.5GHz，整个稳像过程的耗时如下图所示，可以看出，对于每一帧，稳像过程仅需要 20 多毫秒即可完成。在一般的播放器中，每一帧大约需要 30 毫秒来播放，可以预测，再经过一些算法上的改进，整个稳像过程甚至可以做到低延迟实时稳像。

Performance	
Stage	Time / frame
Video decoding (ffmpeg, luma only)	7.91 ms
Equirect to cube conversion	0.73 ms
Pyramid construction	0.87 ms
Feature generation (at key frames)	0.10 ms
Translational Lucas-Kanade tracking	2.19 ms
Rotation estimation (at key frames)	0.10 ms
Pure-rotation optimization (inner frames)	2.20 ms
Deformed-rot optimization (inner frames)	1.84 ms
Warp coordinate computation	3.46 ms
Frame warping	2.20 ms
Total	21.60 ms

Frame duration: < 33.00ms

图 3.32 混合 3D-2D 算法的运行时间（每一帧）

除了运行时间，该算法在降低比特率方面也有着很好的效果，下图分析了以 H.264/MPEG-4 AVC 格式编码的视频，设置相同的质量参数时比特率的降低。可以看出经过稳像算法处理后，比特率的降低基本是稳定的，大约在 10% 到 20% 中间。图中还可以看出较为平缓的视频（右图）得到的结果更加稳定，可能是由于经过稳像过程，场景变得近乎静止，存在冗余的信息。



图 3.33 稳像算法带来的比特率的降低

该算法有着很好的稳像效果，但是仍然存在着一些缺陷。比如当面对比较强的滚动快门变形的时候，尤其是频率大于帧速率的高频变形时，该算法的“变形-旋转”模型不能很好地表示这种变形，这样的话，这些变形不会被完全消除掉，使得最终的结果有些许瑕疵。此外，“变形-旋转”模型在实际的运行中偶尔会引入一些轻微的摆动，这种摆动相对平滑，在成果视频中也能够观察到。最后，对于一些包含标志或是静态文字的视频，该稳像算法在稳定抖动的背景时，会使这些静态的标志或文字产生摇摆。

延时摄影算法

由于 Facebook 的新算法使得全景视频看起来较为平滑，所以也可以用来创建加速延时视频。创建一个延时摄影 360 视频需要删除完整拍摄内容的一部分，并把每帧序列连接起来。然而，延时摄影拍摄与普通拍摄的一个共同点是需要平稳的平衡相机速率。例如，在拍摄一个滑雪视频时，滑雪者时而加快速度，时而停下休息，相机速率也在不断变化。为了模拟一个不断移动的延时摄影镜头，便需要暂时地平衡速度，并跳过休息的部分。

要做到这一点，首先用二维近似和平均运动矢量，估计每一帧的相机速率。然后，利用时间中位数和低通滤波器对视频进行二次处理。通过相机预估速率，就可以改变原视频的时间戳。这样一来，就可以创建加速视频，把冗长的视频缩短。

Facebook 表示，这项新算法还在测试当中，希望能够得到用户的反馈。用户现在可以试着上传 360 度视频，把高质量，且观感更舒适的视频与大家分享。接下来，Facebook 将致力于改进延时摄影算法，希望在不久的将来，能用到实时 360 视频中。

小结

全景视频是现阶段 VR 的核心、本质，但仍存在着一些问题和难关。除了分辨率不足的大方向，还有就是现如今大多数的全景视频都是固定拍摄的，给人们的体验有些乏味。针对提升分辨率的方向，很多研究者都在着手于对高分辨率全景视频流的编码、传输等方面的研究；而对于运动条件拍摄的全景视频，会引起强烈的抖动是限制其广泛传播的一大原因。Facebook Research 的 Johannes Koef 提出了一种针对与全景视频稳像的混合的 3D-2D 算法，有着不错的效果。这也给了我们一些启示和今后的研究方向，值得我们学习。

本章参考资料：

[1] http://www.360doc.com/content/18/0508/10/5109282_752096194.shtml

- [2]<https://wenku.baidu.com/view/fe7f934b0622192e453610661ed9ad51f01d5485.html>
- [3]<http://baijiahao.baidu.com/s?id=1599885711498451402&wfr=spider&for=pc>
- [4]<https://docs.unrealengine.com/en-us/Engine/MediaFramework/HowTo/FileMediaSource>
- [5]<https://blog.csdn.net/caozhaodan/article/details/77647847>
- [6]M. Grundmann, V. Kwatra, I. Essa, Auto-directed video stabilization with robust L1 optimal camera paths[C], IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2011: 225–232.
- [7]H. Qu, Li Song, Video stabilization with L1 – L2 optimization[C], IEEE International Conference on Image Processing (ICIP). 2013: 29–33.
- [8]Accelerometer/gyro-facilitated video stabilization, Apple Patent, US8558903, 2013. 10
- [9]J. Kopf, 360° video stabilization [J]. ACM Transactions on Graphics (TOG), 2016, 35(6) :195.
- [10]<https://code.facebook.com/posts/697469023742261/360-video-stabilization-a-new-algorithm-for-smoother-360-video-viewing/>
- [11]<http://www.ifanr.com/644385>
- [12]<https://blog.csdn.net/u013816144/article/details/53635299>
- [13]<https://www.sciencedirect.com/science/article/pii/S0262885615000311>

第四章 全景视频流媒体技术

4.1 MPEG OMAF

由于全景视频需要较大的视频分辨率(4K或者8K,甚至16K),必然会导致媒体数据量的剧增。而如何提高对巨大的数据量的压缩效率,如何在延时较低的情况下对全景视频进行传输,这些都是全景多媒体应用对传统架构方案的挑战。目前,市场上出现的虚拟现实产品参差不齐,标准不一,造成了一定的行业混乱,需要新的行业标准的约束。因此,为了将VR技术扩展到更广泛的市场,需要定义一种通用的应用架构标准,可以在不同的VR设备之间进行VR视频的存储,管理,交换,编辑和呈现。

4.1.1 全景媒体应用的发展与演进

全景媒体的应用格式(OMAF)最先由MPEG组织在2015年10月的日内瓦举行的113届MPEG会议上提出,它提出的重要意义在于它为VR系统的输入输出接口设定了标准,使得VR技术可以以一种更加规范的姿态扩展到科学的研究和商业领域。与传统媒体应用格式相比,全景视频从捕获到播放是一种端到端的技术,由于视频分辨率太高,因此在过程中,容易形成视频内容“切片化”,而MPEG组织建立OMAF标准,正是为了避免全景媒体内容的碎片化。首先,OMAF框架可用于将360度视频与二维视频图像相互转换的映射和渲染;其次,在ISO基本媒体文件格式(ISOBMFF)的基础上,框架的文件存储模块扩充和丰富了VR视频存储功能和相关信令的定义;此外,在框架还增加了能支持基于流媒体协议的动态自适应流的封装和传输;并且它针对全景媒体流提出了更高要求的压缩编码性能。

在全景多媒体应用格式的概念提出之初,对其应用架构的需求也随之而出。首先,OMAF的系统架构需要支持多种媒体内容的播放和存储和全景视频的压缩,同时也兼容已有的文件格式,传输系统和编解码器。在2016年2月,Byeongdoo Choi等人提出要增加对2D/3D音频编码的要求和3D音频和视频之间的空间同步,以及支持基于用户视角的编码、传递和渲染处理的需求。到2017年4月的第118届MPEG会议时,已经形成了较完善的全景媒体应用的需求规范,包括了对传输、视觉质量、音频格式以及安全性的需求。



图 4.1 OMAF 标准草案的发展阶段

MPEG组织在提出OMAF的标准之时,就初步形成了较为粗略的应用框图,涵盖了图像拼接和映射、视频编解码以及视频在球面的渲染模块。Youngkwon Lim在初始框图的基础上,详细的规范了模块之间的I/O接口的规定形式,促进了VR系统开发的一致性。2016年6月,Byeongdoo Choi等人在各种MPEG会议的提案基础上,对OMAF可支持的映射方式进行了初步的总结,罗列了近十种映射方式,并进行了比较和归纳。在编码方面,J. Ridge等人提出了下一代VR编码的趋势,对编解码器的实际应用和部署都提出了挑战性的需求。在存储方面,Ye-Kui Wang等人提出了一个支持存储全景媒体内容的文件格式的标准,在原有的MPEG文件格式中,增加了多个关联VR的BOX类型。在传输方面,Franck Denoual等人提出了一个新的DASH描述符来帮助DASH客户端利用和管理VR内容,并且提出使用SRD来进行基于用户视角的流式传输。如图4.1所示为OMAF标准草案的一个发展阶段路线图,从目前的研

究趋势可以看出，从 2017 年 1 月开始就有成规范体系的关于 OMAF 框架的 MPEG 会议输出文档(CDs)。

如图 4.2 所示，是关于 MPEG OMAF 的未来发展路线图。可以看到，在 2017 年底，三自由度的全景 VR 系统架构的标准制定完毕，到 2020 年底，六自由度全景 VR 系统也将会发布，届时对编解码、传输、文件格式等有全新的标准支持。

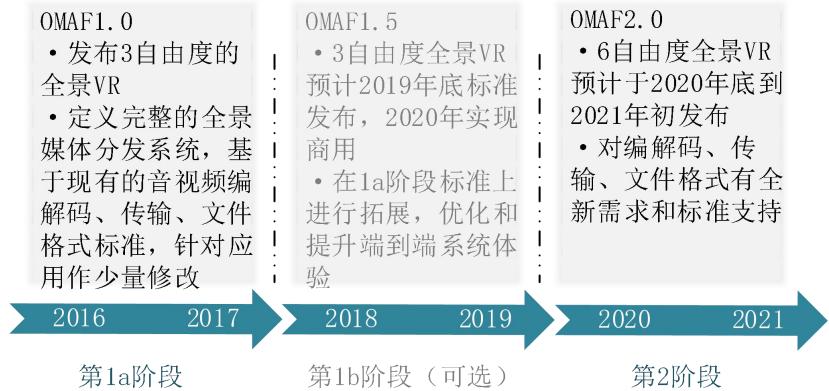


图 4.2 MPEG 组织的 OMAF 标准未来发展路线图

关于全景媒体应用的架构，国内标准组织 AVS 在 2015 年下半年也启动了 VR 全景视频应用工作计划，其任务和目标是着重围绕着视频编码，定义全景视频紧凑表示方法和编码工具，促进 VR 设备互联互通，提升全景视频压缩效率。在第一阶段（2015 年底至 2017 年 3 月），AVS 组织致力于研究全景视频编码与现有平面视频编码标准的兼容性；在第二阶段（2017 年 3 月至 2018 年 3 月），AVS 组织将重点放在定义新的全景 3D 视频编码工具上；在第三阶段（2017 年 3 月至 2020 年 3 月）实现六自由度全景视频的编码。

IEEE 虚拟现实与增强现实标准工作组旗下正在制定的八项标准(IEEE P.2048)，其中涉及到全景多媒体架构的包括：沉浸式视频分类和质量标准和沉浸式视频文件和流格式这两个标准。截至 2017 年 4 月，全球共有接近 200 个企业和机构的专家参与该标准的制定工作，成为 VR/AR 标准化的主要推动力量。

4.1.2 全景媒体的应用架构

与传统 2D 媒体应用架构不同，全景媒体的应用架构所处理的对象为数据量较大的 2D/3D 的全景图片、全景视频和 3D 音频等。因此，全景媒体架构对于传统模块如编解码、封装、传输等提出了更高的性能要求。同时在应用需求上，为了将 3D 媒体内容到 2D 平面之间的相互转换，还需要映射和渲染模块的支持；由于全景媒体特有的交互性特点，观看者视角这一元素也须考虑进入整体架构中。这些性能与应用上的需求构成了全景媒体架构的关键元素，基于这些思想，各大组织和企业在研究和发展中，形成了逐渐完善的更为细化的架构。

4.1.2.1 MPEG OMAF 下的全景应用框架

MPEG 组织自 2015 年底开始建立 OMAF 标准的目的是为了避免全景媒体内容的碎片化。为了支持全景媒体的应用，基于 OMAF 的全景媒体的系统框架应运而生，流程图如下图所示。

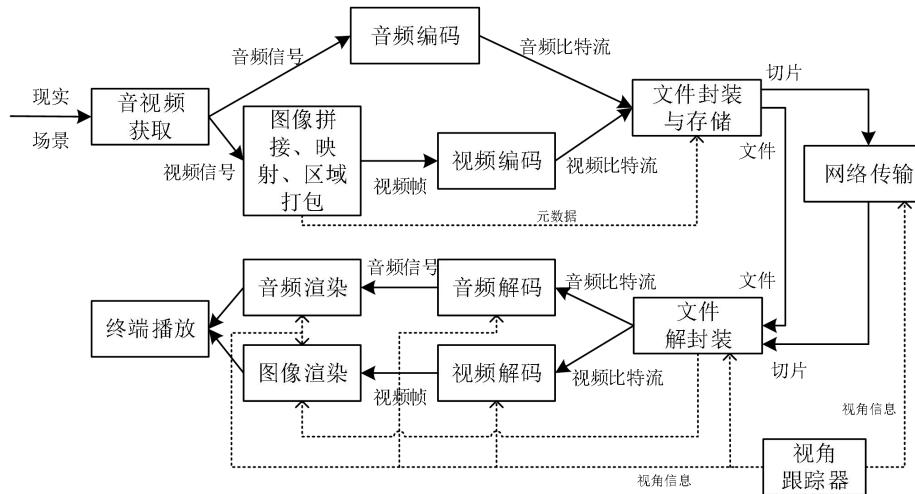


图 4.3 OMAF 下的全景媒体应用框架流程图

图 4.3 中, 实线表示音视频的数据流向, 虚线表示 OMAF 和用户视角的信令流向。可以看到, 不同于传统媒体应用框架, 在客户端的模块均受视角信令的控制, 体现出 OMAF 框架的用户交互性; 另一方面, 全景媒体的映射等信息也通过 OMAF 信令在封装和渲染模块之间传递。对于全景媒体应用的各个模块, MPEG 组织经过不断的研究和讨论, 提出了丰富的解决思路和算法。

4.1.2.1.1 全景媒体的获取

对于全景视频的采集，理论上可以通过全光函数来实现。全光函数是一个从空间中任意点 (V_x, V_y, V_z) 以任意角度 (θ, φ) 在任何时刻 t 所看到的任意波长 λ 的光线集合的函数，它的定义如下式所示。

$$P_7 = P(\theta, \varphi, V_x, V_y, V_z, \lambda, t) \quad (4.1)$$

而全光函数所要求的信息量过大，采集、传输和显示等技术问题短期难以获得突破。全景视频是全光函数的近似，它将 7 维表达简化到 4 维，固定了位置，仅靠球面的视角 (θ, φ) 、入射光的波长 λ 以及时刻 t 来表达观看的场景。早期的全景成像系统使用的是兼反折射的摄像机，它由单镜头相机和反光镜组成，利用反光镜，将周围的图像信息反射到相机上的成像面，采集到相机平面整个半球面域内的图像信息。但由于其结构特点，在相机顶部会存在盲区，无法捕获高质量高分辨率的全景视频。目前，较为常用的全景媒体获取的方法是使用包含具有重叠视野的多个鱼眼相机组成的系统。

在 MPEG 第 115 次会议上，高通和 LG 提出鱼眼相机视频相关的文件格式语法和语义，在 MPEG 的 OMAF 标准中，鱼眼相机的两个有关拼接和渲染的参数，光学变形校正（LDC）和镜头阴影补偿（LSC），被纳入了 OMAF 的信令中以提高图像渲染的质量。所谓光学变形，则是指鱼眼镜头成像存在失真，主要体现在空间点在成像面上的实际像点跟理论上的像点之间存在误差，通常使用基于球面透视投影约束的校正算法来消除光学变形。此外，镜头阴影是在多个图像拼接时在缝合处产生的黑色阴影，它严重影响了视频质量，目前阴影补偿的常用方法包括线性相关补偿法和信息补偿法等。

4.1.2.1.2 映射格式

全景视频经过相机组的获取和拼接后，还原出一个 360 度的球面视频。然而，目前的视

频编解码器都需要 2D 图像的输入，因此，拼接好的球面视频需要经过映射模型来完成三维到二维的变换，下图是以经纬图映射模型（ERP）为例说明了映射变换的原理。经过多次探讨和决策后，目前 OMAF 仅支持 equirectangular (ERP) 和 cubemap 投影。

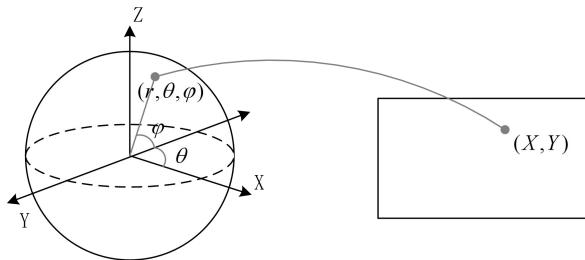


图 4.4 映射变换示意图（经纬图模型）

4.1.2.1.3 编解码方案

目前，市场上主流的编码方案是将球面全景视频通过映射转换为的平面视频使用传统平面视频编码的方法进行视频压缩。传统视频编码技术对全景视频仍然有效，新一代视频 (H.266/FVC) 预期能适度减少一半视频码率，可部分缓解全景视频传输的带宽压力。然而，不同于传统视频，全景视频有其画面的独特性，针对这些特点，一些创新的编码优化思想应运而生。Madhukar Budagavi 在 2015 年提出在编码之前加入区域自适应的平滑模块，平滑由经纬图模型映射导致过采样的两极区域像素，平滑后节省大量码率。Guoxin Jin 等人同年提出了新的扭曲运动补偿方法来解决由于鱼眼镜头拍摄造成的运动变形。MPEG 组织也在 2015 年提出了适应于全景视频编解码的需求：(1) 针对不同镜头和映射方式均实现较好的压缩效率；(2) 减少相机组透镜之间的冗余进一步提升压缩效率；(3) 编解码器能从压缩比特流中提取视角区域流；(4) 编解码器应存储于光学有关的校正和预处理参数，能在渲染端准确再现场景。

在 2017 年 4 月，MPEG 组织提出了适用于全景视频的八个可能性的编解码方案。传统方案是使用时间帧间预测 (TIP) 将视频图像编码为单层比特流，并传送到接收机侧，由解码器完全解码，将当前视角区域呈现给用户。传统方案不受视角信息的控制，感兴趣区域和背景区域的编码方法完全一致。基于视角编码的方案有子图像序列法、感兴趣区域增强层法、同分辨率 HEVC 序列法、不同分辨率 HEVC 序列法等。子图像比特流方法是将全景映射图像在编码之前被分割成子图像序列，每个子序列可以独立编码为不同比特率的比特流组合，在解码端根据当前视角区域选择相应序列的不同比特率版本。感兴趣区域增强层法是分别对整个映射图像（底层）编码以及对分块图像进行不同比特率编码，编码之后传送到解码端，解码底层映射图像以及当前观看方向（感兴趣区域）的图像并呈现。

基于 HEVC 运动约束分块集 (MCTS) 法是将 HEVC 流按照相同分辨率进行不同质量（假设红色为高质量，黑色为低质量）和比特率的编码，如下图所示。在接收端，在当前视角区域选择质量高的分块，背景区域选择质量低的分块分别解码后进行组合产生混合质量的图像。

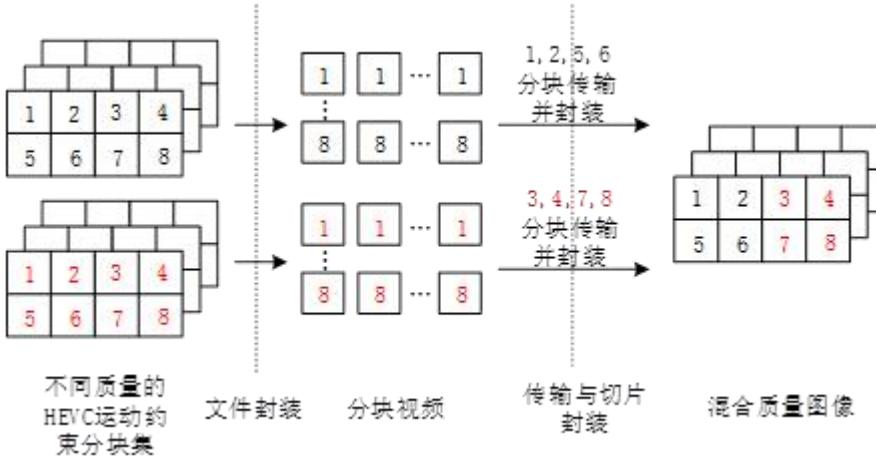


图 4.5 相同分辨率 HEVC 序列法流程图

除此之外，还有不同分辨率的 MCTS 序列法、可缩放编码的部分解码 (SLPD) 等基于视角的全景视频编解码方案。基于视角的全景视频编解码能够在固定带宽情况下，根据观看质量合理的分配码率，将当前视角区域的视频分辨率和质量提升而降低背景区域的质量，有效的契合了全景视频的特点，提升了终端用户的观看体验。

4.1.2.1.4 传输机制

传统的传输机制缺乏对全景内容的支持，例如，客户端不知道媒体流片段为全景媒体，那在传输中，全景视频的超大分辨率对于带宽和实时性的要求便提出了高难度的挑战。在全景视频中，同一时刻，观众只能观看某一视角内容，基于这一特点，若根据用户视角进行动态切换主视点码流，即采用动态流切换方式，则能去除“视角”冗余，减少带宽压力。在 OMAF 标准中，提出了两套传输方案：DASH (HTTP 动态自适应流媒体) 方案和 MMT (MPEG 媒体传输) 方案。

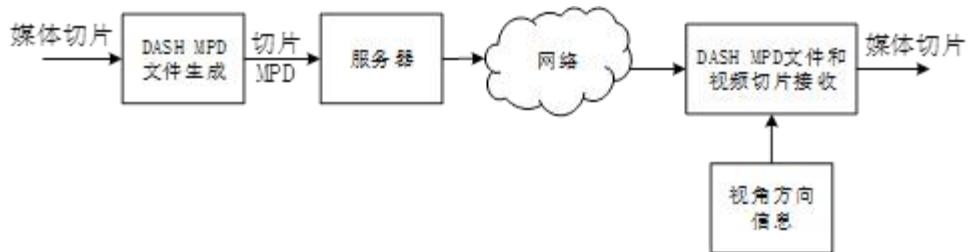


图 4.6 基于视角的 DASH 流程图

应用于 OMAF 中的 DASH 方案传承 DASH 基本思想，它通过牺牲存储空间来提高带宽利用率。每个视角都存储多份不同码率的视频流，同一时刻传输主视角的较高码率和其它视角的较低码率，相当于码率和视角构成的“二维 DASH”协议。如图中所示，在传统的 DASH 机制基础上，增加了用户视角信息，使得视角信息从客户端反馈到服务端从而进行动态码流的切换。在传输系统设计中，需要兼并权衡存储、带宽节省、延时等各因素最大化用户体验和空间、带宽利用率。并且 OMAF 添加了一些新的描述符：(1) All under the URN “urn:mpeg:mpegI:omaf:2017”；(2) Projection format (PF) descriptor；(3) Projection format (PF) descriptor；(4) Content coverage (CC) descriptor；(5) Spherical region-wise quality ranking (SRQR) descriptor；(6) 2D region-wise quality ranking (2DQR) descriptor；(7) Fisheye omnidirectional video (FOMV) descriptor。

基于分块以及视角切换等思想，传输方案的设计和编解码方案一脉相承。例如在图 4.5

中表示的 HEVC 运动约束分块集 (MCTS) 法，在编码端将全景图像划分为多个分块，称为 tile，每个 tile 又会有不同的码率，然后编码为不同的码流，根据用户视角信息在网络传输中动态切换不同 tile 和码流的媒体流，并在解码端组合成高质量主视角和低质量背景的混合图像。而且 DASH 协议中也在之前就提出了 srd(spatial relationship descriptor) 的概念，也正是为空间分 tile 做好了准备，它可以定义每一个 tile 在接收端要呈现的位置，好进一步能够使得播放器能够认识每个 tile 在播放的时候需要呈现的位置是哪里。

另一种传输方案 MMT 也可作为 OMAF 应用架构传输模块的候选。它的流程图如下图所示。

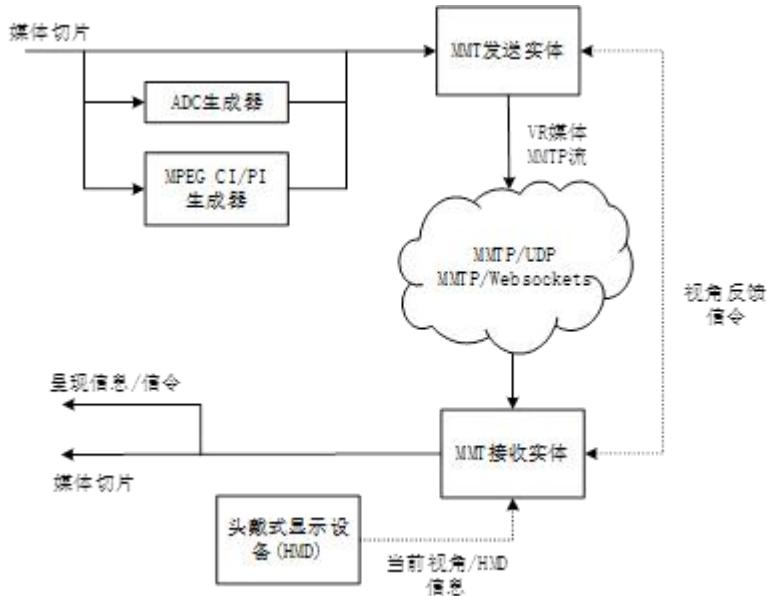


图 4.7 基于视角的 MMT 流程图

MMT 和 DASH 同是 MPEG 组织标准下的传输协议，二者除了传输架构不同之外，在 OMAF 架构下，MMT 与 DASH 方案相同，在全景视频的传输中，需要根据当前视角方向，传递全景视频的主视角流，可以根据客户端指定当前视口，也可由发送端的服务器来选择。

4.1.2.1.5 存储格式

为了在 ISOBMFF 中支持 OMAF 作为媒体存储和封装格式的应用，ISOBMFF 内部本身需要进行一些 BOX 类型的扩展。在 OMAF 制定和完善的进程中，关于 OMAF 的存储格式尚未完全达成一致标准。目前的主流实现方案是在 ISOBMFF 文件的基础上增加多个视频轨道，并在轨道层次上，添加更多 VR 信息来支持 OMAF 这一格式。

为了在 track 层次来表达 VR 视频的信息，Ye-Kui Wang 等人提出后解码需求机制，它是通过对受限方案信息 box (Restricted Scheme Information box) 中信息的添加和修改来加以实现的。由于在语义层对 ISOBMFF 标准进行了拓展和修改，因此常规的解码器和播放器并不能对 OMAF 的文件格式进行正常解析和播放。因此，为了处理 VR 视频对播放器或渲染器上在操作的一些需求，后解码需求机制要求播放器具有能够简单检查文件来找出渲染视频比特流的能力。

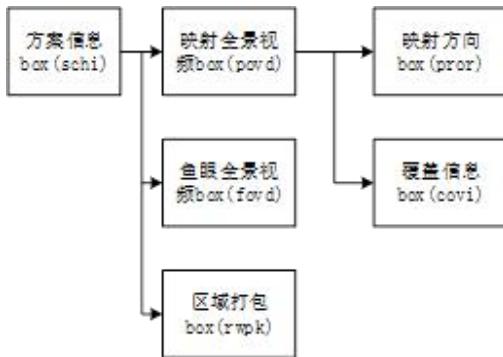


图 4.8 OMAF 在受限视频方案中的新增信息

修正的 OMAF 标准里，将受限视频方案(Restricted video schemes)纳入了正式文档中，在原有的 ISOBMFF 标准基础上，加入了映射、打包和鱼眼视频等相关 BOX 的表达，有关 OMAF 的新增信息如图 4.8 所示。其中，映射全景视频 box (povd) 中有许多映射方式的信息；鱼眼全景视频 box (fovd) 包含了鱼眼全景视频的参数信息；区域打包 box (rwpk) 表示图像映射后通过了区域打包模块，并且需要在渲染之前进行解包处理。在 povd 内部，映射方向 box (pror) 包含了映射模型的坐标系方向信息，覆盖信息 box (covi) 表示全景球体表面的相关信息。

下图显示了 OMAF 目前支持的 9 种多媒体文件：

- 3 video profiles
 - HEVC-based viewport-independent OMAF video profile
 - HEVC-based viewport-dependent OMAF video profile
 - AVC-based viewport-dependent OMAF video profile
- 2 audio profile
 - OMAF 3D audio baseline profile
 - OMAF 2D audio legacy profile
- 2 image profiles
 - OMAF HEVC image profile
 - OMAF legacy image profile
- 2 timed text profiles
 - OMAF IMSC1 timed text profile
 - OMAF WebVTT timed text profile

图 4.9 OMAF 支持的 9 种多媒体

除了加入受限视频方案(Restricted video schemes)，在 OMAF 的文件存储格式领域，还有一些目前正在研究的技术，例如，子图像的多路 track 技术，将分块的基于视角的多路子视频由多个 track 来描述，改变了原有的文件格式多数情况下只有一路视频 track 的情形。此外还有区域视频质量排名技术，它可以用子评定在同一轨道的其他区域或者其他轨道之间的质量优劣。

4.1.2.2 主流公司的全景媒体应用架构

VR 技术是目前最受关注的前沿科技之一，受到了各家互联网公司的青睐，但这并不是首次。实际上，VR 在发展史上经历了三次热潮。第一次热潮发生在上个世纪 60 年代，出现了第一个计算机图像驱动的头戴式显示设备以及头部位置跟踪系统，是 VR 发展历史上的一个重要里程碑。第二次热潮发生在上个世纪 90 年代，3D 游戏的上市使得 VR 技术关注度剧

增，但由于当时 VR 技术尚不成熟，游戏画质差价格高，因而这一次的 VR 热潮就此消退。到 2014 年，Facebook 公司收购 Oculus 后，VR 热潮再度袭来，Facebook 创始人在中国发展高层论坛中说道 2016 年将成为消费者 VR 之年，并且，在 2017 年 4 月底的 Facebook F8 大会上，Facebook 甚至表示未来 VR 设备可以直接取代智能手机。目前，越来越多的大型科技公司开始涉足 VR 领域。

4.1.2.2.1 Facebook 全景媒体应用方案

无论是 VR 社交还是 VR 游戏，这些仅仅是 Facebook 的 VR 展现形式而已，而支撑 VR 应用的核心是一个支持全景媒体的数字通信架构。这与 MPEG OMAF 架构类似，Facebook 在全景媒体的应用架构中的从媒体获取到渲染播放端的关键技术如下图所示。

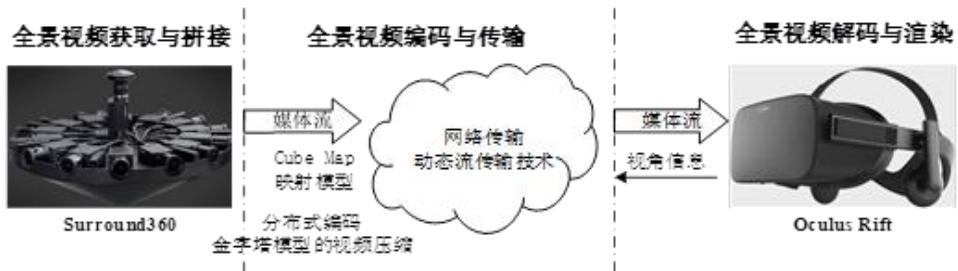


图 4.10 Facebook 的全景视频关键技术示意图

基于 Surround360 相机的视频获取与拼接

全景媒体应用框架的输入是对 360 视频的获取模块，Facebook 在 2016 年发布了 Surround360 摄像机，并且将硬件设计和图像拼接代码开源到网上。Surround360 是一个高品质的 3D 全景视频采集系统，可以生成真正的球面 VR 效果，并且内部配有拼接软件，大大减少了后期制作的难度。Surround360 由环绕 360 度的 17 台摄像机组成，它将拍摄到的多路视频进行拼接并将其转换成适合于 VR 观看的立体 360 全景。

对于一个 360 度的视频，它在拼接时存在很多传统 2D 视频没有的困难，比如，多路摄像机产生的海量数据处理，人眼视觉对 3D 视频拼接的错误的低容忍度，以及运用到实践中所要求的处理时间效率。在拼接模块，Surround360 采用了基于光流的算法，用光流来计算左右眼立体视差，对左眼和右眼分别合成对应视角方向的虚拟摄像机的新视图，然后再将左右眼的视图重新组合。这种方法可以捕捉场景的运动，以达到远胜于普通拼接的无缝立体效果。拼接后的输出为每只眼睛提供 4K、6K 和 8K 视频，其中 8K 视频已经超过业界的标准输出，保证了最佳的观看体验。内部的拼接系统也节省了后期制作时间，在效率上提供了保障。

正六面体映射方式

Surround360 将多路相机拍摄到的视频以经纬图映射的方法为输出，而对于 360 度视频，如果用这种传统的映射方式来呈现，则会在顶上与底下两部分包含较多的冗余信息，且呈现效果较为扭曲，不符合人眼视觉习惯。Facebook 在映射方式上选择了正六面体的方法，将经纬图的布局重新映射到正六面体上，正六面体是六面正方形的集合，属于视角独立的映射格式。

正六面体映射方法有很多的优点，比如在立方体的每一个面上没有任何失真，每一面的映射都是相对独立的。其次，视频编解码器中运动矢量为直线形式，正六面体不会像经纬图方法那样将图像扭曲，因此这种映射方式对编解码器非常友好。此外，它的像素点分布较为均匀，不包含冗余信息。在 Facebook 的方案中，为了实现从经纬图方法为显示到立方体映射的转换，它创建了一个自定义的视频过滤器，使用多点投影的方式来进行二者之间的像素

点切换。这套方案通过将经纬图视频的顶部的 25% 转换为一个立方体面，将底部的 25% 转换为另一个立方体面，中间的 50% 转换为四个立方体面。这样，正六面体的输出包含与经纬图输入相同的信息，但每帧的像素数量减少了 25%，提高了空间的效率。

基于分布式编码与金字塔型视频压缩

在编码方面，为了在合理的时间内处理海量数据的全景视频，Facebook 使用了分布式编码，在多台机器上编码不同的视频分块，并随后将其接近无损的拼接在一起。另一方面，Facebook 采用金字塔模型压缩算法，能使得全景视频文件无损压缩到原来的 20%。金字塔模型是一个与视角相关的立体映射模型，它的底部为用户视角区域的全分辨率视频，随着金字塔高度的上升，在金字塔其他面上的视频质量逐渐变低，压缩率逐渐增加。而当用户切换视角时，并不是给用户看该金字塔其他表面的低质量视频，而是切换另一个以下一视角为底部的金字塔模型。在 Facebook 的方案中，一个经纬图的输出将被转换为 30 个视角的金字塔模型，基本能覆盖整个全景视频的各个视角空间。每一个金字塔有五种不同的分辨率版本，因此，对于一个全景视频，一共有 150 个不同版本的编码流。在后续的传输中，这些视频都预先被存储在服务器上，虽然这耗费了大量空间，但不需对每个客户端的请求进行实时编码，因此降低了用户在视角之间切换时的延时，保证了观看质量和效果。

动态流传输技术

在全景视频的传输方面，Facebook 在 2016 年 1 月提出了动态流技术。由于有限的网络带宽与计算能力的限制，传递超大数据规模的全景视频会造成缓冲或者中断等问题。Facebook 针对这些难题，与 MPEG OMAF 的思想类似，提出了基于视角的自适应比特流技术，在视觉感兴趣的区域提供最高质量的视频，同时降低外围背景的视频质量，因此它在缩小比特率的同时，保证视角区域的观看质量。在客户端对于下一个视频块的选择，针对目前的网络条件以及综合分辨率，视频质量，当前视角方向等元素，可以考虑数十种不同的可能的流来呈现。其次，在传输中服务端需要频繁的更新网络状况，以更短的时间估计网络带宽，这样能保证系统能做到及时调整，避免缓冲延迟的发生。此外，在 DASH 通过 HTTP 传输自适应流时，流通常包含两个特定块：初始化块和索引块。其中，初始化块包含为每个媒体块添加的编解码器的初始化数据，索引块包含搜索映射和表示中每个块的确切字节范围数据。如果要切换到新流，这两个块的信息是必需的。Facebook 在传输方案上，在 DASH 的 list 中为所有动态流媒体流在后台预取这两者，因此，只要播放器需要切换到新的流，就无需花费时间来重新获取，这样提高了时间效率。

然而，若服务器端不知道用户的当前视角方向，如何进行自适应流的切换呢？Facebook 基于这种情形开发了基于内容的动态流技术，它主要是依靠人工智能（AI）给出的显著图数据来实现的，它利用显著图的统计数据计算出观看者可能的关注点和兴趣点。在处理完视频的每一帧后，客户端会收到一个单个流的视频，它在感兴趣区域提供高质量，而无需用户去选择码流，所以被称为是基于内容的流技术。

与基于视角的流传输技术，基于内容的码流传输技术有以下优势：首先在功能上，它可以支持流的缓冲、下载和离线播放；其次，它允许长视频段或者更多的关键帧被一次性传送，从而降低比特率并改善压缩；由于不需要用户切换流，所以它没有分辨率的跳转，从而简化播放。

目前，对于全景视频的动态流传输技术已经成功的运用在了多个厂家的 VR 设备上。

Oculus Rift 头戴式显示设备

Oculus Rift 是 Facebook 目前的主流头显产品，它以较大的视场角和较高的分辨率的

优势减少了画面延迟和避免了晕动症。目前，已经有多款应用和游戏登陆 Oculus Rift。而 Facebook 创始人称，Oculus Rift 将从“沉浸式游戏”开始，最终扩张到其他的体验平台，比如远程教育或者活动的“现场”体验。此外，凭借 Facebook 广阔的社交平台，很有可能将会开启一个数字交流的新时代。

4.1.2.2 Google 全景媒体应用方案

在 2014 年，Google 推出了一款价格低廉的 DIY 设备 Cardboard，实现所有的手机都可以变身“VR 查看器”，它与 Facebook 旗下的 Oculus 推出的高端 VR 设备形成了鲜明的对比。而如今，白手起家的 Google 已经凭借其各种高端交互式设备站到了 VR 和 AR 技术领域的最前沿。

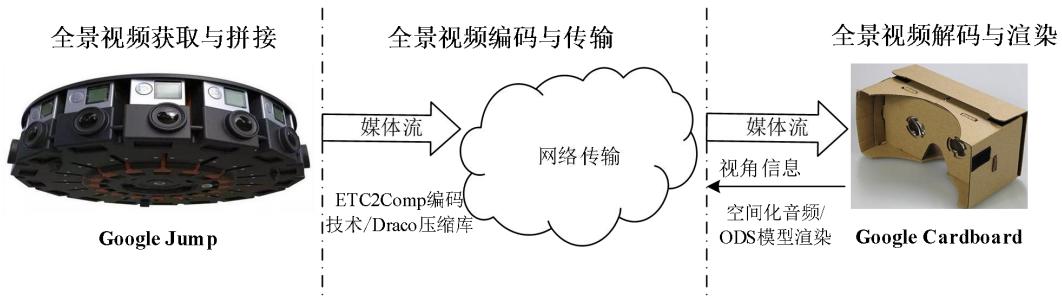


图 4.11 Google 全景媒体关键技术

Google Jump 全景摄像装置

在全景媒体应用框架的输入口，Google 推出了 Google Jump，它是由 16 台 GoPro 镜头组、自动拼接软件和播放平台 3 部分组成。Jump 拍摄的原始视频经过 JUMP 应用转换后，会生成非常逼真的 3D 虚拟现实视频。严格来讲，Jump 是 Google 虚拟现实视频内容制作的设备，它最重要的意义在于降低虚拟现实内容制作和消费的门槛，让虚拟现实变得触手可及。

ETC2Comp 编码技术与 Draco 压缩库

在编码压缩这一环节，Google 发布了 ETC2Comp 技术，它是一款用于游戏和 VR 开发的编码器。在编码 360 视频的过程中，ETC2Comp 通过部署一些优化技术，可以以更快的速度获得高质量的视觉效果。在优化策略中，ETC2Comp 通过“定向块”的搜索方式有针对性的获得给定块的最佳编码方式，这种压缩方式可以比使用暴力法快得多。在代码方面，由于每个视频块可以进行独立编码，ETC2Comp 采用了高度多线程。此外，Chrome Media 团队创建了 Draco，这是一个开源的压缩库，用于改善 3D 图像的存储和传输性能。Draco 压缩库提高了 3D 图像的压缩效率而不会影响视觉保真度。对于用户端来说，下载的速度更快，在浏览器中的 3D 图像也可以更快的加载，并且减小了 VR 场景的带宽传输压力，使得全景内容能快速呈现。

空间化音频技术

除了对全景视频方面的处理，Google VR 团队还在整个全景媒体框架中引入了空间化音频技术，通过将空间音频引入网页，浏览器可以转换成一个完整的 VR 媒体播放器。如图 4.12 所示，用户可以听到浏览器上的 360 度环绕声。解码器能记录包含 4 通道的音频，然后将其解码成任意的扬声器设置。此外，使用 8 个虚拟扬声器，代替实际的扬声器阵列，以双向呈现最终音频流。这种双耳呈现的音频可以在通过耳机听到时传达空间感，为在网络上更加沉浸式的 VR 体验发挥了关键作用。

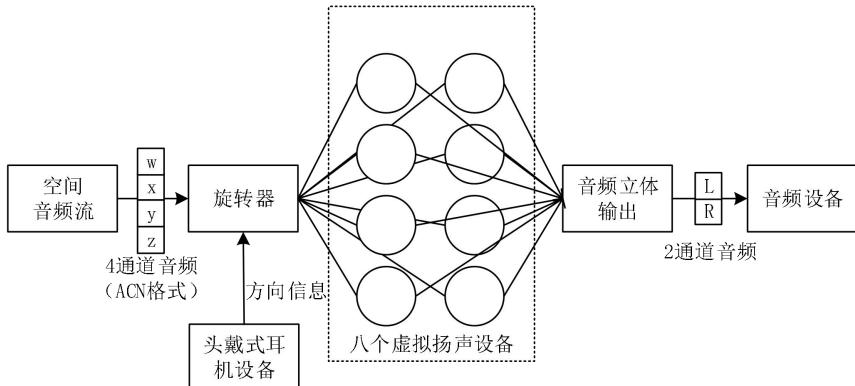


图 4.12 空间化音频技术的流程图

基于 ODS 模型的渲染技术

在渲染方面，Google 设计了一个映射模型 ODS (Omni-directional stereo) 在头显设备上渲染全景视频，这个映射模型只捕获每个摄像机的中心射线，并借用其他摄像机的其他射线方向，如图 4.13 所示。所以这种光线捕获方法呈现在左右眼的射线几何将会变得更加的立体和全方位。ODS 都是采用的预渲染方式，在头戴式设备中播放都非常的流畅。

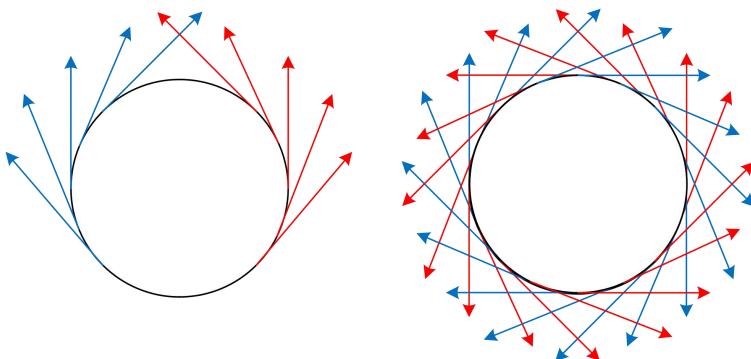


图 4.13 基于 ODS 模型的光线捕获模式

虽然 Google 加入 VR 领域迟于 Facebook，但发展势头很猛，相比于 Facebook 有自己社交平台的优势，Google 也致力于以 Android 操作系统为核心的 VR 应用，比如 Daydream，同时也在推行一体化 VR 头盔，即不再需要连接智能手机或电脑。不像 Facebook 收购 Oculus 公司开发 Rift 头戴式设备，Google 从硬件到生产到内容输出正在构建闭环系统和框架，在前后端继续研究全景媒体应用领域。

4.1.3 全景媒体应用架构的对比

在上一节中，我们介绍了在 MPEG 标准下的 OMAF 框架、Facebook 和 Google 的全景媒体应用架构，它们都针对数据量庞大的全景视频进行了端到端的处理。以下我们在架构严谨性、架构发展阶段以及行业标准上对这几种框架进行对比。

在结构严谨性上，MPEG 的 OMAF 标准中，在各个模块环节如映射、编码、传输等上都提出了不同的解决方案，例如八种基于视角的视频编码以及传输方案。此外，还进行了对比性的实验，具有多种尚在研究中的全景媒体架构的选择方案。其次，OMAF 在框架结构上也更加的具体和严谨，同时在很多细节上都进行了完善和优化，例如映射后图像进行基于区域的打包等。在这一点上，Facebook 和 Google 在框架中的大多数模块中都没有平行方案。

从发展阶段上而言，Facebook 和 Google 这两个具有代表性的科技公司，主要都是以产

品为核心，技术为支持的产业路线，从产品的端到端的需求和性能来布局全景媒体架构，在系统实现中去优化相应关键技术点，使系统更加完善，更加符合市场需求。而目前，MPEG 下的 OMAF 框架中各个模块的细节和采用的算法还在进一步的研究和讨论中，因此并未形成行业内的标准，还停留在实验阶段，没有投入产业使用。同样，参与制定虚拟现实行业标准的 AVS 组织也还在其第二阶段的标准制定中。

从行业标准上而言，由于 2016 年虚拟现实行业热潮爆发，各种 VR 产品参差不齐，导致行业标准混乱，为了避免 VR 市场分散，需要全景媒体框架的标准化，而 MPEG OMAF 正是制定整个虚拟现实的行业标准，它的领头性是不容小觑的。但是，这一标准并不限制日新月异的 VR 市场创新和少数产品的各异性。而 Facebook 和 Google 作为创新性公司，其架构与 OMAF 框架存在差异，但并不影响其后续的发展与延伸。同一市场中的不同分支总是求同存异，谋求交叉发展。

总的来说，全景媒体应用框架的讨论、制定和完善是一个具有挑战性的课题。8K 甚至更大分辨率的全景视频对于网络带宽提出了高难度的需求。随着全景媒体直播技术的发展，延迟将是影响用户体验的一个重要参数，而终端显示设备播放的视频质量也决定了用户观看效果。这些关键技术的研究将对今后虚拟现实技术的发展具有十分重要的研究意义和应用价值。尽管目前 VR 技术在游戏、社交等领域发展迅速，但它内部的系统结构仍需要继续细化和完善，全景多媒体应用的发展还处于起步阶段。如今，越来越多的组织和企业都加入到制定 VR 行业标准的队伍中，提供了新的思考和方法，因此值得展开更为深入的研究。

4.2 HEVC, DASH 等相关扩展

在之前章节中提到过，编解码与流媒体传输也是全景视频呈现中较为重要的一部分，接下来将对这两部分目前流行的标准、技术做简要的介绍。

HEVC

高效视频编码（High Efficiency Video Coding，简称 HEVC），又称为 H.265 和 MPEG-H 第 2 部分，是一种视频压缩标准，被视为是 ITU-T H.264/MPEG-4 AVC 标准的继任者。2004 年由 ISO/IEC Moving Picture Experts Group (MPEG) 和 ITU-T Video Coding Experts Group (VCEG) 作为 ISO/IEC 23008-2 MPEG-H Part 2 或称作 ITU-T H.265 开始制定。第一版的 HEVC/H.265 视频压缩标准在 2013 年 4 月 13 日被接受为国际电信联盟 (ITU-T) 的正式标准。HEVC 被认为不仅提升视频品质，同时也能达到 H.264/MPEG-4 AVC 两倍之压缩率（等同于同样画面品质下比特率减少到了 50%），可支持 4K 分辨率甚至到超高清电视 (UHDTV)，最高分辨率可达到 8192×4320 (8K 分辨率)。HEVC 技术对于移动互联网应用的首要意义在于，移动直播时码率更低、减少对网络的冲击、大幅度节省带宽费用。相比与 H.264，HEVC 在继承它的应用模块下又进行了优化，HEVC 使用预测与变换相互结合的混合视频框架，图 4.14 展示了 HEVC 的编码框架，首先将一帧图像划分为递归四叉树结构，接着进行帧内预测与帧间预测，得到一个预测图像块，将预测图像块与原图像相减得到残差块，然后依次对残差块进行 DCT 变换、量化与熵编码，最终得到压缩后的视频码流。

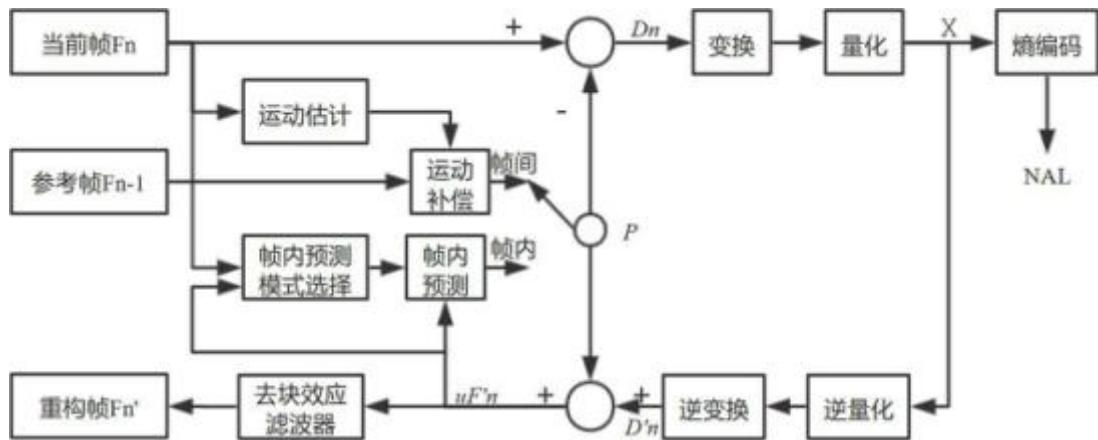


图 4.14 HEVC 编码框架

- 图像分块：在 HEVC 标准中，使用宏块划分图像，将宏块的大小从 H.264 的 16×16 扩展到了 64×64 ，以便于高分辨率视频的压缩。同时，采用了更加灵活的编码结构来提高编码效率，包括编码单元 (CodingUnit)、预测单元 (PredictUnit) 和变换单元 (TransformUnit)。编码单元类似于 H.264/AVC 中的宏块的概念，用于编码的过程。预测单元是进行预测的基本单元，变换单元是进行变换和量化的基本单元。这三个单元的分离，使得变换、预测和编码各个处理环节更加灵活，也有利于各环节的划分更加符合视频图像的纹理特征，有利于各个单元更优化的完成各自的功能。
- 预测编码：HEVC 依旧根据视频与空间相关性对视频进行帧内与帧间预测，对于相互关联的相邻像素，通过帧内预测降低空间冗余度，对于残差较小的连续视频帧，使用帧间预测减少时间冗余度，帧内预测上 HEVC 是在 H.264 的预测方向基础上增加了更多的预测方向。
- HEVC 对于所有尺寸的 CU 块，亮度有 35 种预测方向，色度有 5 种预测方向。而 H.264 对于 4×4 的块亮度有 9 个方向， 8×8 块有 9 个方向， 16×16 块有 4 种方向，色度有 4 种方向。HEVC 增加了广义 B 帧预测方式代替 H.264 中的 p 帧预测方式，增加了运动估计的准确度，提高编码效率，有利于编码流程统一。
- 帧间预测：本质上 HEVC 是在 H.264 基础上增加插值的抽头系数个数，改变抽头系数值以及增加运动矢量预测值的候选个数，以达到减少预测残差的目的。HEVC 与 H.264 一样插值精度都是亮度到 $1/4$ ，色度到 $1/8$ 精度，但插值滤波器抽头长度和系数不同。对于帧间预测，HEVC 可以以更高的精度对运动矢量进行编码，从而以更少的残差提供更好的预测块。

视频运营中最大的支出成本就是宽带，对于高分辨率的全景 360 视频来说更是如此。采用新型高效的视频压缩标准将大幅降低全景视频的带宽成本。HEVC 作为目前广泛采用且压缩性能较好的编码标准，兼容全景视频的（分块、分层）传输概念，是现有条件下最适合沉浸式媒体的编码器。实际上，一些新型编码器如 VP9、AV1 也正在研发、优化中，可以期待在未来一段时间内，有更好的编码方式为我们带来高质量、低延迟的沉浸式体验。

流媒体传输技术

流媒体自适应传输是当前流媒体技术领域研究的一个重要方面，特别是本世纪以来，随着互联网技术和移动通信的迅速发展，视频及多媒体信息的网络传输问题成为了信息化过程中的热门问题，尤其是对于高质量视频服务的强烈需求。流媒体技术是指在传送数据的时候采取流式传输，传统意义上的流媒体技术如 RTSP 基于 TCP 协议而 RTP 是基于 UDP 协议的，

而且对防火墙不友好，同时需要配套的专用网络设施。随着 HTTP 额外的带宽开销问题影响越来越小，一些公司如微软、苹果和奥多比公司开始开发基于 HTTP 的新一代流媒体协议。这些协议可以直接利用现有的 CDN (Content Delivery Network)，而且不需要服务器来维护会话状态。但是每个公司的方案都是不开源的，应用上各有利弊，互不兼容。在这样的背景下，由国际标准化组织运动图像专家组 (MPEG) 牵头，以 3GPP 和 OpenIPTVForum 部分内容为基础与 2010 年开始进行标准化工作。2011 年 1 月出台国际标准草案，同年 11 月，MPEG-DASH 成为动态自适应流媒体技术的国际标准。

传统流媒体传输技术

流媒体传输本质上就是基于固定协议的 IP 数据流传输，传统流媒体技术以基于 TCP 协议的实时流协议 (Real Time Streaming Protocol, RTSP) 和基于 UDP 协议的实时传输协议 (Real-time Transport Protocol, RTP)/实时传输控制协议 (Real-time Transport Control Protocol, RTCP) 为主。

RTSP 用于创建和控制终端之间的媒体会话，可以实时控制从服务器到客户端和从客户端到服务器双向的媒体信息。RTP 由 IETF 的多媒体传输工作小组 RTP 由 IETF 的多媒体传输工作小组 1996 年在 RFC 1889 中公布，既可以用在单播也可以进行多播。一个 RTP 分组由 RTP header 和 RTP payload 两部分组成。RTP header 有序列号字段和时间戳来控制视频的播放，payload 里就是具体的视频数据，因为不同的音视频编码标准而不同，如 H.264 编码。RTP 是真正的实时传输协议，客户端仅需要很小的缓冲空间来存储一些参考帧数据，延时可以控制在一秒以内，当网络拥塞时会丢弃一些不那么重要的包保证视频可以流畅播放下去，这也是现在商业上大部分直播选择 RTP 协议的原因。但是 RTP/RTSP 协议需要特殊的网络配套设施，对防火墙不友好，也不能利用现有的 CDN 设备，成本较高。

HTTP 渐进式下载可以很好地利用 HTTP 设施，和把文件完全下载下来不同，渐进式下载可以在缓冲一部分数据之后就进行播放，但是带宽容易浪费，仅适用于点播内容，缺乏灵活的控制机制，不能根据实际环境对播放视频的质量进行选择。而 RTP/RTSP 又对现有的网络设施和防火墙不友好，所以一些基于 HTTP 的动态自适应流媒体传输技术得到应用，如微软的 MSS (Microsoft Smooth Streaming)，奥多比公司的 HDS (HTTP Dynamic Streaming) 以及苹果公司的 HLS (HTTP Live Streaming)，它们都会把源视频切割分块，同时生成索引文件 (Manifest File)，里面是多媒体文件的位置以及相应的时间戳和编码等信息。每个视频分块的时间长度相同，在视频编码层面，意味着每个分块都要包含一个或多个完整的图像群组 (GOP, Group of pictures)，以 I 帧开始，便于独立解码。这些视频分块都被存储在 HTTP Web 服务器中，播放器会检测连接带宽和客户端一些资源的使用情况，进而选择不同参数的视频分片。各个分片间没有重复和不连续，因此在用户看来就是平滑连续的播放。

MSS 是微软公司研发的关于动态自适应流媒体传输的协议，MSS 文件切片为 mp4，索引文件为 xml 格式的 ism/ismc，对直播和点播都支持比较好。ism 是服务器配置文件，用来描述服务器上不同码率视频分片的关系，ismc 是媒体描述文件，指定在某些情况下如何选择分片，对应于合适的码率和分辨率。通过文件级别的 moov 元数据描述视频分片的信息，但是有效的载荷是包含在片段盒子里的片段级别元数据 moof 和 mdat，关闭文件的盒子包含 mfra 报头，可以在整个文件中被精确快速找到。一个典型的平滑流文件片段长度为两秒，可保证时延较低。作为 IIS (Internet InformationService) 的媒体服务扩展，MSS 协议的应用只能依托于 Silverlight 终端技术，当 Silverlight 客户端发出请求时，服务器需要准确地分析 URL 参数，将其转换为对应的文件偏移量，从而定位目标数据位置并作为响应内容回应客户端的请求。MSS 的实现依赖于微软的解决方案，在部署上局限比较大。

HDS 是奥多比公司在流媒体自适应传输领域提出的方案，支持点播和直播两种工作模

式，是 Adobe 给 RTMP 协议的补充，为 Adobe Flash 以及 AIR 客户端提供了基于 HTTP 协议的动态码流切换功能。同时它在视频编码方面支持 H.264 和 VP6，音频编码方面支持 AAC 和 MP3。协议的实现和应用依托于 Adobe 自家的 FMS (Flash Media Server)，具有动态缓存和流加密功能。流媒体索引为 manifest 文件，点播的时候通过 File Packager 将多媒体文件分割并写入 f4f 文件，这种文件格式允许通过 HTTP 协议定位内部分片并进行下载。直播的时候则是通过 Live Packager 实时收集 RTMP 流并将其转化为 f4f 文件，然后将这些文件部署在 Apache 等 web 服务器上，经过封装处理后再转发给播放组件，直播数据流通过 RTMP 协议进行传输，需要服务器相关的配置支持。

MPEG-DASH

基于 HTTP 的动态自适应流（英语：Dynamic Adaptive Streaming over HTTP，缩写 DASH，也称 MPEG-DASH）是一种自适应比特率流技术，使高质量流媒体可以通过传统的 HTTP 网络服务器以互联网传递。类似苹果公司的 HTTP Live Streaming (HLS) 方案，MPEG-DASH 会将内容分解成一系列小型的基于 HTTP 的文件片段，每个片段包含很短长度的可播放内容，而内容总长度可能长达数小时（例如电影或体育赛事直播）。内容将被制成多种比特率的备选片段，以提供多种比特率的版本供选用。当内容被 MPEG-DASH 客户端回放时，客户端将根据当前网络条件自动选择下载和播放哪一个备选方案。客户端将选择可及时下载的最高比特率片段进行播放，从而避免播放卡顿或重新缓冲事件。也因此，MPEG-DASH 客户端可以无缝适应不断变化的网络条件并提供高质量的播放体验，拥有更少的卡顿与重新缓冲发生率。

MPEG-DASH 是一种基于 HTTP 协议进行数据传输的动态流媒体自适应技术，已经成为包括 3GPP、EBU、HBBTV 等多个国际标准的推荐流传输协议。与已有的采用 RTP 的方法相比，HTTP 不需要考虑防火墙的问题，并且可以充分利用已有的系统架构，如缓存、CDN 等。DASH 本身也可以通过 WebSocket 和上层 push 等技术来支持低延迟的流推送，而且不同于 HLS、HDS 和 Smooth Streaming，DASH 不关心编解码器，因此它可以接受任何编码格式编码的内容，如 HEVC、H.264、VP9 等。由于其多方面的优势，目前全景视频也主要采用 DASH 协议进行传输。

MPEG-DASH 协议定义了一个层次化结构的文件架构。首先定义一个 Media Presentation Description (MPD) 文件，MPD 用 XML 语言书写，它包含分片的 HTTP Uniform Resource Locators (URLs)、格式、带宽、分辨率、编码方式等，MPD 的分层结构如下：

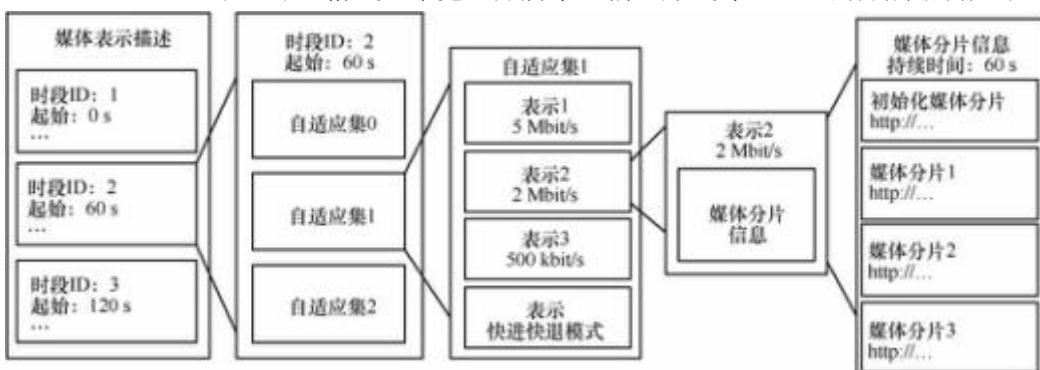


图 4.15 MPEG-DASH 索引文件 MPD 格式

- MPD 文件包含许多组时间连续且独立的时间段集合 (period)，确定视频的起始时间与播放时间。
- 每个时间段 (period) 集合包含同段多媒体内容的多个自适应集合 (Adaptation Set)，自适应集合用来标识区分相互关联却互不相同的多媒体内容，例如同一时段下的视频，音频，字幕。

- 每个自适应集合（Adaptation Set）包含多个媒体内容集合表示（Representation），媒体内容集合表示包含了视频的码率，带宽，编码等信息。
- 媒体内容集合（Representation）包含多个切片（segment）及对应的 URL 信息，切片为实际的多媒体切片文件，可以通过对应的 URL 用 HTTP GET 进行视频下载与播放。

对于 DASH 文件，不同码率分辨率的分片组都会有各自的初始分片，即 init.mp4，用于初始化播放组件。这个视频片段相比于一般包含具体媒体数据的视频分片（例如时长 2s 左右）较小，在 2kb 左右，不存储实际的视频内容，而是包含了解封装所需要的全部元信息，即在 moov 盒子中存储的内容，比如视频流和音频流各自的编码格式和相关参数、分辨率以及时间轴等。视频分片一般都是 m4s 格式的，与之前的初始分片相对应，只是包含媒体信息 (moof+mdat)，通过资源定位符 URLs 进行定位，包含一个或多个 GOP。长一点的分片会使 HTTP 传输更有效率，因为对于每次传输时附加的 HTTP 头信息都是类似的，短一点的分片主要用于直播方面，DASH 传输的实质在于当一个分片完全封装完成时才能作为 HTTP 小文件传输出去，所以这是降低直播时延的有效方法，而且分片较小有利于在网络波动比较大的时候及时调整视频质量，防止播放中断。m4s 切片格式是基于 ISO-BMFF (ISO Base Media File Format, ISO 基础媒体格式) 存在的，这种格式是 MPEG-4 标准中的一部分。ISO-BMFF 文件格式以对象化的方式组织内容，使用一系列盒子（box）来描述各层次包含媒体信息的容器，全部信息都包含在各种特定的盒子中，这些盒子按顺序排列，每个盒子又分层次地包含更次一级的盒子，通过逐层嵌套来描述媒体信息的细节，在功能上分为头结点和数据域两个部分。

流媒体视频播放时，DASH 客户端首先通过 HTTP 下载 MPD 文件，解析文件得到对应的多媒体内容。时间长度、媒体格式、分辨率、带宽限制等信息。根据这些信息，客户端根据网络带宽状况与缓存区存储深度来调整视频码率，向服务端申请对应的视频切片，进行下载与播放。MPEG-DASH 只定义了 MPD 文件格式与视频切片格式，对于数据传输、切片编译码方法和客户端码率选择都没有规定，在流媒体传输系统设计时具有较高的灵活性与拓展性。

4.3 流媒体系统

目前，360 视频的流媒体传输主要有以下几种形式：1) 交互式流媒体，用于视频会议、游戏等场景；2) 现场直播，如体育赛事、演唱会的实时在线播放；3) 流媒体点播，Youtube、Facebook 等网站上的视频播放大多采用这样形式。

不同于传统 2D 视频，360 视频可感知的分辨率范围取决于视角跨度。之前章节中提到过，人眼视网膜可以区分出最高 60 像素每度 (PPD) 的分辨率。一般的 HD 视频具有 36–100 的 PPD。然而，相同分辨率的 360 视频因其大跨度 ($360^\circ \times 180^\circ$) 的需求，PPD 会降至 11 左右，导致用户在观看时会感受到画面模糊的现象。如果将 360 视频的 PPD 同样提升到 60PPD，则其在 HMD 中显示需要 $5400^2 \sim 7200^2$ 个像素点，形成完整全景画面需要约 21600×10800 个像素点，再考虑帧率、色彩等因素，这种理想条件的视频播放会消耗 2.35Gbps 的带宽。

此外，360 视频是在视场 (FoV) 跟踪的基础上进行播放的，因而引入了运动-图像 (MTP) 形式的延迟。作为一种沉浸式的体验，这种延迟不应高于 20ms，具体而言，其图像渲染和传输延迟均不应高于 10ms。在理想状态下，总延迟应降至 10ms 甚至更低。然而在现有的传输环境下，360 视频的平均传输带宽只能达到 18.7Mbps，播放延迟为 80ms，与最终目标还有很大的差距，因而 360 视频流媒体传输对于相关领域的研究人员来说仍是一个巨大的挑战。

目前，高效的 360 视频流传输系统的目标之一是最小化所需的传输带宽，使得视频可以通过家庭有线 DSL 线路或无线连接 (WLAN 或 4G/5G 网络) 快速有效地传送。自适应视频流

传输已成为当前视频传输的标准。自适应流传输主要通过客户端和服务器之间的交互来实现，例如，基于可用带宽和容量，客户端从服务器请求下一个视频片段以匹配网络当前的传输能力。自适应流传输尽可能地避免由于缓冲区欠载导致的客户端播放卡顿以及随后再次缓冲的过程。在符合 MPEG 动态自适应流传输标准 HTTP (DASH) 的流媒体系统中，服务器存储两种类型的文件：

- 1) 媒体呈现描述文件 (MPD)，其包含关于可用片段，内容的 URL 和其他相关信息；
- 2) 实际的视频元数据。

客户端启动流式处理。客户端首先接收 MPD 并解析。通过解析 MPD，客户端了解不同编码流的可用性，它们的位置和其他媒体特性。然后，客户端根据可用带宽请求下一个子段进行流传输。下载的片段被渲染并通过 HMD 显示在屏幕上。

人类视觉系统 (HVS) 只能感知整个 360 度自然空间的一部分。这一点也是设计 HMD 时要考虑的重要因素。例如，Samsung Gear VR 2016 HMD 的最大水平视场 (FoV) 为 101 度。这意味着在任何时刻，用户能看到的范围远小于全向视频内容的水平 360 度 FoV。因此，流媒体系统始终以高质量传输完整的 360°x180° 内容几乎没有价值。

因此，为了有效地节省带宽，可以以高质量传输视角区域的图像，而视野外的背景以较低的信噪比 (SNR) 质量或较低的分辨率进行传输。当然，也可以使用其他一些带宽节省的方案。像这种类型的选择性流传输策略被称为基于视角的传输 (VDD)，与以相同高质量传输完整全景视频的方案相比，VDD 可节省大量带宽。

由于头部运动，用户在观看 360 视频时会有短时间内偏离当时视角的情况。在这种情况下，系统会从背景中向用户呈现较低质量的视频。而在背景渲染的同时，客户端同时请求服务器传输更新后的视角所对应的视频内容。前述操作呈现了视角切换的过程。一旦客户端接收到与新视角相对应的媒体数据，就再次向用户呈现高质量视频。以图 4.16 为例：当用户头部向右转动，超出 (a) 中第一个 (当前) FoV 时，发生视角切换。类似地，向右方向进一步的头部运动会产生如 (c) 和 (d) 中沿着整个 360 度水平空间的视角切换。

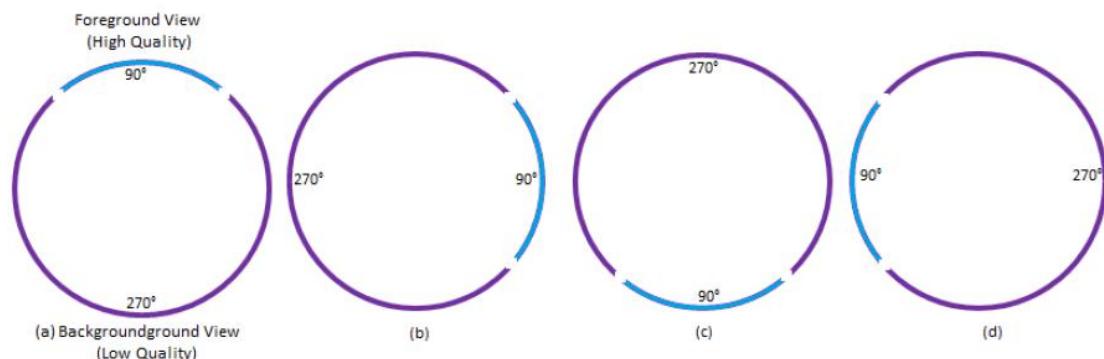


图 4.16 90 度 FoV 的 VDD

VDD 方案的效率在很大程度上取决于两个重要因素：

1) 视角大小：FoV 很大程度上会影响传输带宽和渲染。更广的视角需要更多部分的 360 度视频以高质量进行编码和传送。此外（图 4.17 A），由于 HMD 视角的限制，FoV 内显示的数据可能比高质量数据要少得多。这导致带宽浪费。另一方面，由于头部运动，太小的视角会导致过于频繁的视角切换操作，这可能会对视觉质量产生负面影响（图 4.17 B）。此外，如果系统设定的 FoV 小于 HMD FoV，则可能需要对多个视角进行选择性传输和显示方可完全覆盖 HMD FoV，这可能会导致视角间的边界产生可见的伪像。

2) 运动-高质量图像 (MTHQ) 延迟，是指从头部运动到当前视角外区域开始，至系统反应，并在 HMD 上显示刷新视角产生高质量图像所经过的时间。高 MTHQ 延迟会产生明显的视觉质量损失，应该避免。

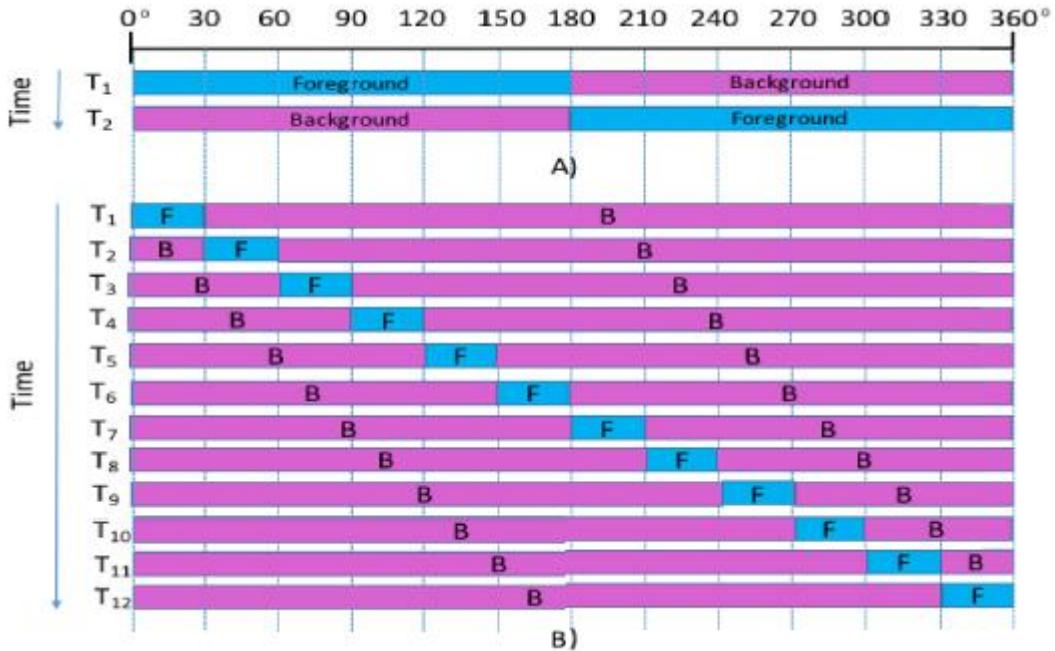


图 4.17 两种情景: A) 大视角情况 B) 小视角情况 (F 表示视角内的范围, B 表示视角外的范围)

基于视角的 VR 传输延迟

考虑完整的 HMD 端到端系统, 以头部运动作为触发, 直至用户在 HMD 中看到更新的高质量图像。MTHQ 延迟由图 4.18 所示因素决定, 主要有以下几种:

- 1) 传感器延迟: 头部移动转化为有效传感器信号的时间。
- 2) 网络请求延迟: 取决于 CDN 边缘与用户的接近度, 因此取决于 CDN 的密度。此延迟会影响所有视角自适应方法。
- 3) 源端-边缘延迟: 图 4.18 中最左边两个模块间延迟的总和。这是由 CDN 中的缓存未命中引起的延迟。
- 4) 最重要的延迟是由于使用 LAN 或 WiFi 时本地 (家庭) 网络的传输或通过接入网络从边缘传输至客户端引起的。根本原因在于往返时间, 可用带宽和请求大小。后两个因素共同形成传输延迟。
- 5) 解码前的缓冲延迟: 从接收比特流片段到将比特流送至解码器的时间。此延迟很大程度上取决于流协议和打包格式。
- 6) 解码延迟: 由随机访问等待时间和解码器流水线的设计决定。
- 7) HMD 渲染延迟: 取决于操作系统的帧缓冲架构。
- 8) 解码视角的大小同样影响 MTHQ 延迟。

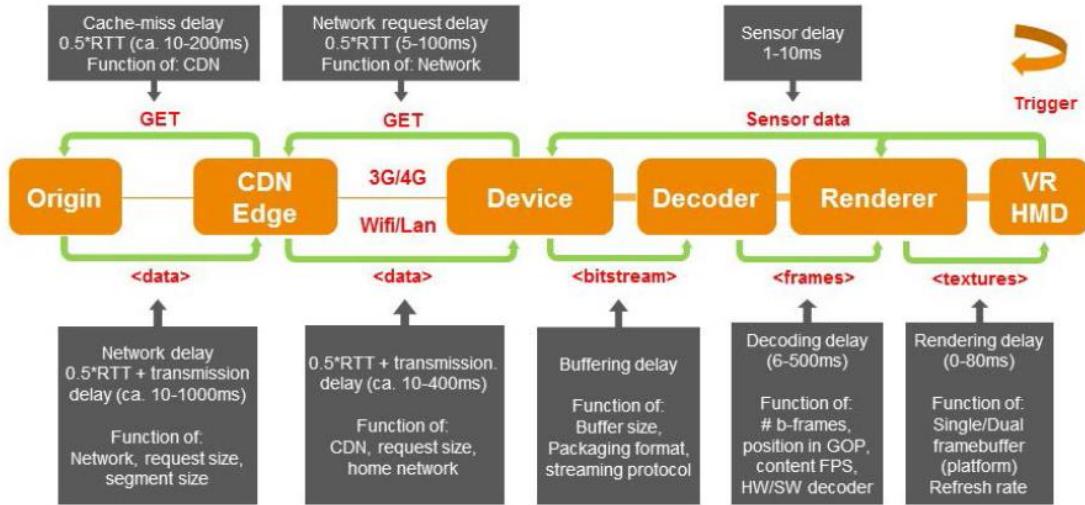


图 4.18 VR 系统中的延迟因素

4.3.1 简单的原型系统

目前有多种 VDD 方法可以将系统带宽降低到符合实际的水平。最主要以下两种：

- 1) 创建许多不同版本的全景视频并在它们之间切换，具体取决于用户的注视方向；
- 2) 将图像划分为图像块 tile，仅传输视角内的 tile 或依据视角分层传输 tile。

实际上，目前第二种方法是一种更具前瞻性和灵活性的选项，可扩展性更高，需要更少的编码和服务器资源。

基于 tile 的流媒体传输优势

基于 tile 的流媒体传输有以下的一些优点，使该方法非常适用于商业服务。

- 1) 质量：基于 tile 的流媒体满足同时使用可供的比特率并以非常高的质量流式传输 VR 内容。使用该方法可以降低多达 70% 的比特率而不会降低质量，或相比传统方法，在相同比特率条件下显著提高图像质量。
- 2) 可靠分层：由于低质量基础层始终存在，因此用户不会遇到“黑屏”或卡顿的情况。此外，由于 tile 是独立检索的，当可用带宽降低时，视角中心可能仍然能显示高质量的图块，而视角角落会降低质量显示。比特流自适应技术的使用进一步确保了用户可以在任何时间点，任何可用比特率条件下体验最佳质量。
- 3) 低延迟：基本层的 MTP 延迟几乎为零，并且在有利的网络（CDN）条件下，可以在一帧或两帧内检索到高质量 tile，这使得质量切换现象不太明显。

4) 编码器兼容：基于 tile 的 VR 流与现有的行业标准编解码器兼容，尤其是 HEVC 编解码器。

5) 解码器的高效使用：基于 tile 的 VR 传输过程中会利用一侧新出现的 tile 替换从视角另一侧消失的 tile，从而实现带宽控制。如果保持带宽不变，该方案可以防止临时的质量下降。

6) 设备支持：除头戴式设备外，基于 tile 的流媒体还可与“平面屏幕”配合使用，例如平板电脑，手机，甚至电视。

7) 支持直播：与其他方法相比，该方案得到完整的 VR 全景只需编码一次。基于 tile 的 VR 流媒体除了适用于点播外，还非常适合直播服务。与需要进行最多 30 次（每个视角一次）编码的其他方法相比，这使得基于 tile 的 VR 成为直播/点播的可扩展且易于部署的解决方案。

8) 可扩展性：基于 tile 的流媒体使用与所有商业部署的流媒体服务相同的标准 HTTP 概念。这意味着标准服务器或 CDN 无需进行任何更改即可将此类 VR 流部署到顶层或托管网络上，并同时分发至大量用户。

依据基于 tile 的视频分块思想，本节将介绍一种简化的原型系统的搭建过程。在简化系统的基础上，便可以对视频以不同分辨率（或质量等）进行编码、传输，并依照用户视角等信息，在终端提供两种甚至多种分辨率混合而成的视频，如图 4.19 所示。

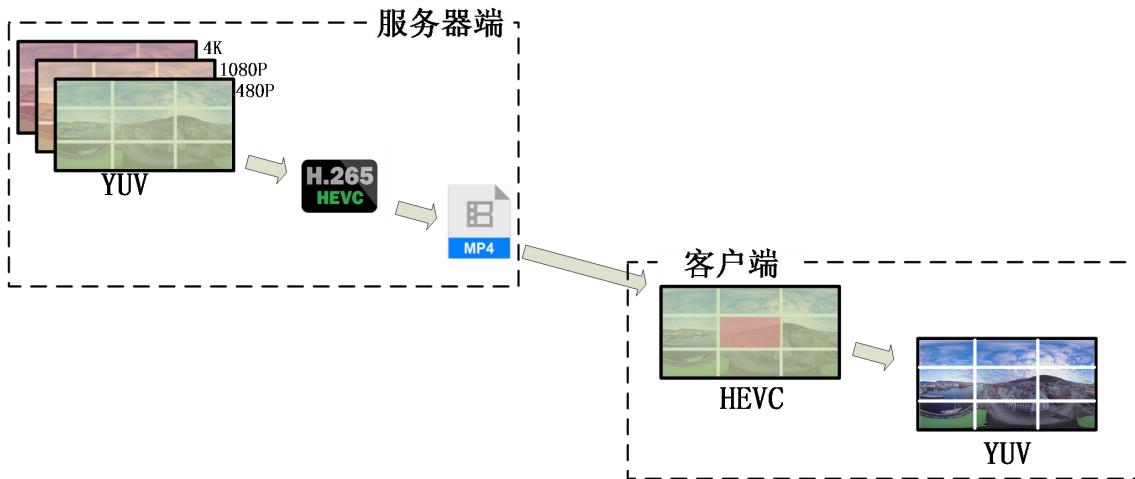


图 4.19 多分辨率分块传输系统

原型系统主要分为服务器端和客户端，其中包含编码器、数据封装模块、数据解封装模块、动态解码器等多个子模块。

在服务器端，系统获取经纬图投影后的多种分辨率视频作为输入，为 YUV 格式。为实现对用户视角区域的动态处理，原始 YUV 码流经编码器后形成空间划分，得到 HEVC 码流，再经封装后生成可传输的 MP4 文件。形成具有划分的封装后，客户端便可以根据用户视角请求获得任意一个 tile。

对于客户端而言，当服务器端具有不同分辨率的视频帧时，其会请求获取所需数据并生成完整的虚拟视图。对于基于 tile 的系统而言，客户端的任务是为虚拟视野区域检索高分辨率的 tile，周围区域不检索或检索质量较差的 tile。关于多分辨率（质量）检索或选择的具体方法，还将 4.4 节中展开描述。最后，客户端依照抽取出的 tile 顺序和视频帧序列进行解封装、解码等步骤，生成中间的 HEVC 流与最终可播放的 YUV 流。因此，与全景高分辨率视频传输方法相比，该途径能够为用户提供高质量的虚拟现实体验，同时节省带宽。

服务器端

目前已有的 GPAC 系统可以很好地支持服务器端的编码及封装过程。GPAC 系统可以解决多媒体领域的诸多关键问题。主要包括：

- 1、多媒体系统架构：多媒体服务的创作，传送和呈现；
- 2、多媒体信息的可扩展编码和适应性；
- 3、多媒体信息安全；
- 4、分布式多媒体服务。

传输系统则主要运用到 GPAC 在视频编码方面的灵活性。实现对视频内容的动态处理不仅需要在编码流中形成空间划分，同时应在封装格式中形成对应的多通道结构，方便码流的抽取。而 GPAC 系统集合了 HEVC 标准下的 kvazaar 编码和 MP4 Box 工具箱，同时支持数据的

分块、分通道传输。

kvazaar 编码是一种学术型开源视频编码器，采用 HEVC/H. 265 标准，模块化的源代码便于多核处理器上的并行化以及硬件上的算法加速。kvazaar 可在帧率、分辨率、tile 分块形式等众多参数设定下对 YUV 格式视频进行分块编码，输出 HEVC 码流，效果如图 4.20 所示。



图 4.20 分块 (8x4) 编码

由于基于 tile 的动态解码流程需要随机抽取解码，须在编码流中插入比例较大的 I 帧，因而对于原始 YUV 码流可以设置每十帧为一组进行编码，使得 I 帧间隔为 10 帧，即为最小的传输单元。kvazaar 编码不仅在编码方式上灵活多变，编码效率也非常可观。

GPAC 系统下的 MP4 Box 主要提供视频封装功能。此工具箱可检测到 HEVC 码流中的视频分块情况，包括分块数量，分块顺序等，在此基础上将各个 tile 的数据流分布至独立的数据通道内，形成编码流与封装的数据通道一一对应，即数据存储顺序以 tile 为大结构，对于某位置的 tile 数据，按照帧顺序存储。最终多个通道的码流并入一个 MP4 文件，形成封装文件。图 4.21 为利用 MP4 Explorer 软件解析到的数据结构，经上述方法封装后，数据被分入多个数据通道 (track) 内。

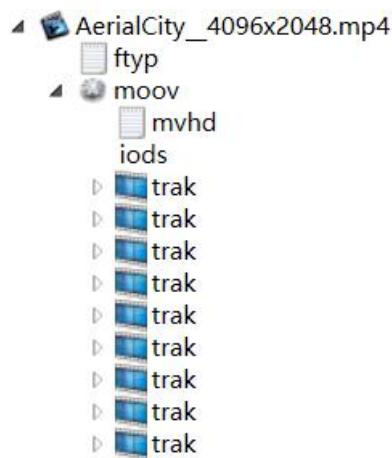


图 4.21 多通道封装

其中，第一路通道为 base tile track，包含 HEVC 文件的参数集信息以及 SEI 信息。其余通道是 tile 数据通道，包含了每个 tile 的数据。此外，MP4 Box 提供通道删除功能，效果如图 4.22 所示，封装文件中去除了一个固定位置 tile 对应的数据通道。这一功能允许服务器端在客户端信令下仅提供所需的 tile 作为后续的传输输入，可以有效减少传输数据量。



图 4.22 MP4 Box 通道删除功能

多通道封装结构也契合 OMAF 的基本思想，将 VR 超大视频编码后统一存储到单个 MP4 文件中，通过通道顺序去索引对应的图像编码流，有利于动态抽取，同时避免 VR 视频在空间上分块后产生许多小文件，从而不易于存储管理。

经过上述过程得到的 MP4 文件即是服务器端的输出，在本节的原型端到端传输系统中，将直接送入客户端进行解封装等终端操作。

客户端

原型系统客户端在接收到 MP4 文件后进行图像块选择、解封装、码流提取、解码与裁剪共四部分操作，最终生成基于用户视角动态显示的 YUV 序列。

1) 图像块选择

图像块选择作为终端视角自适应处理的首个步骤，需明确用户当前视角内的 tile 数量、序号等信息。图像块选择中，首先要说明视角 FoV 信息文件，该文件中涉及到的参数有：FoV 分辨率，每一帧 FoV 的左上角坐标，如表 4.1 所示。其中，FOV 分辨率根据人眼以及 HMD 的可视范围计算得到。此外，图像块的选择以及后续系统的执行还需利用到完整区域的分辨率与分块方式的参数，如表 4.2 所示，这里以 4K 视频为例。结合以上两种数据，系统便可以得到用户视角在整个区域的位置。

表 4.1 FOV 配置文件列表

FoV 配置参数	参数设置
FoV 区域分辨率大小	4K:1180x960 (对应于 110° x90°)
FoV 左上角坐标位置	坐标对：(x ₁ ,y ₁)

表 4.2 解码配置文件列表

解码配置参数	参数设置
源视频分辨率大小	4K:4096x2048
tile 划分方式	8x4

其次，需要明确视频完整区域与 tile 顺序、位置的映射关系。MP4 Box 在封装时以 Z 字线的方式获取 tile，因而以 8x4 分块方案为例，tile 序号如图 4.23 分布。

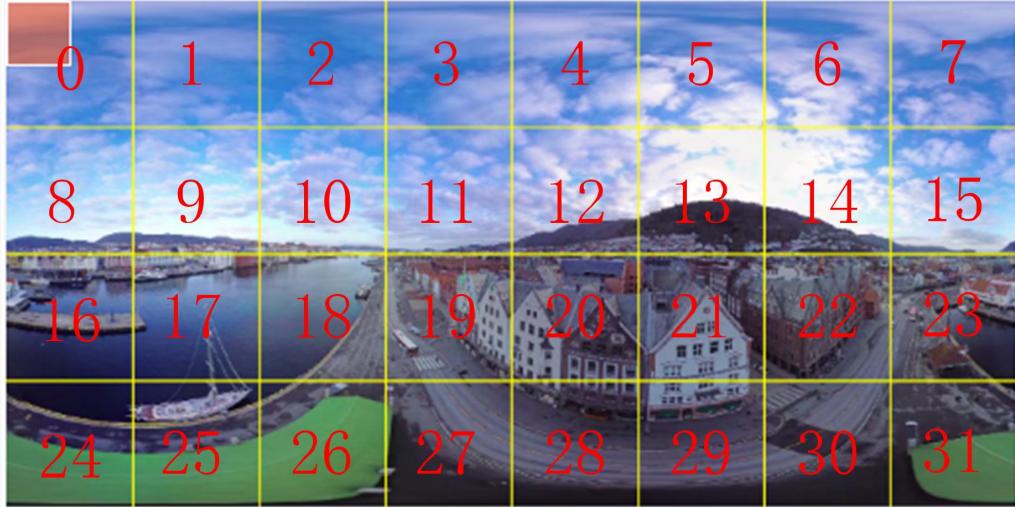


图 4.23 tile 顺序

由于 kvazaar 是均匀分块编码的，因而各个 tile 的长 l_{tile} x 宽 w_{tile} = (视频长度/图像块列数) x (视频宽度/图像块行数)，进而各个 tile 的位置也可以确定。在得到 FoV 位置与 tile 的序号、位置等信息后，便可以确定存在于当前用户 FoV 区域内的 tile 序号集。为使用户可以在视野内看到完整场景，所要提取的 tile 集合应覆盖 FoV 区域，满足如下关系：

$$\begin{cases} x_2 = x_1 + l_{FOV} \\ y_2 = y_1 + w_{FOV} \\ x_1 \geq c_{\min} * l_{tile} \\ x_2 \leq (c_{\max} + 1) * l_{tile} \\ y_1 \geq r_{\min} * w_{tile} \\ y_2 \leq (r_{\max} + 1) * w_{tile} \end{cases} \quad (4.2)$$

其中， (x_1, y_1) 是 FoV 左上角坐标， (x_2, y_2) 是 FoV 右下角坐标， l_{FOV}, w_{FOV} 是 FoV 的长度和宽度， c_{\min} ， c_{\max} 分别是所需 tile 占据的最小和最大列序号(从 0 开始)， r_{\min} ， r_{\max} 则分别是所需 tile 占据的最小和最大行序号(同样从 0 开始)。为降低传输带宽，tile 占据的行列序号最值应取满足条件的最苛刻值。

得到 tile 所占行列序号后，可以通过简单计算得到如下的 tile 集合：

$$\{N_{tile} \mid N_{tile} = r * n_{columns} + c, c_{\min} \leq c \leq c_{\max}, r_{\min} \leq r \leq r_{\max}\} \quad (4.3)$$

其中， N_{tile} 代表 tile 序号， $n_{columns}$ 代表 tile 列数。

再以图形化的样式介绍该过程。这里以 4K 视频为例，如图 4.24 所示，整个 4K 视频帧被划分为 8x4 个 tile，因而每一个 tile 的大小为 512x512，假定 HMD 的视野范围为 110° x 90°，则 4k 视频 FoV 的分辨率大小约为 1250x1024。假设人眼目前所看区域左上角位于 0 号 tile 内，通过式 (4.2)、式 (4.3) 便可以计算得到为涵盖整个 FoV 区域所需的最少 tile 序号为 (0, 1, 2, 8, 9, 10, 16, 17, 18)，即图中蓝色部分。

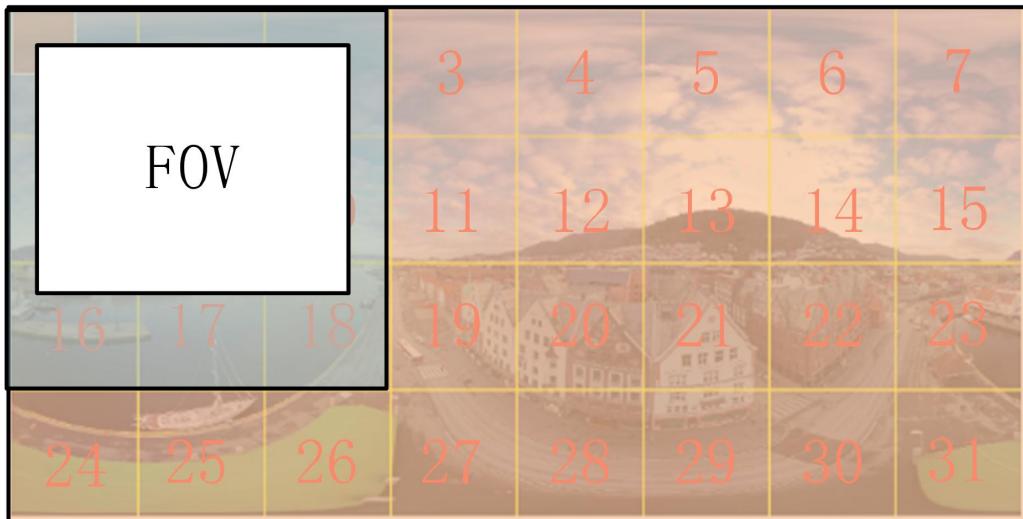


图 4.24 FoV 区域与 tile 的映射关系

通过这种方法，客户端便可以根据 FoV 走向，即对于给定的 FoV 区域分辨率和每一时刻的 FoV 左上角位置坐标，确定该时刻需要抽取的 tile 序号即数据通道序号，此序号集作为信令传入服务器端指示封装模块的通道删除操作，减少不必要的数据传输，同时作为后续码流提取与拼接模块的提示信息。

2) 解封装

在服务器端接收序号集信令并传输相应 MP4 文件至客户端后，客户端需对 MP4 文件进行解封装，方可进行 HEVC 和 YUV 层面的视频数据处理。MP4 文件由大量数据盒 box 组成，不同类型的 box 存放不同类型的数据。MP4 主要有 ftyp、mdat 和 moov 三大类 box。MP4 头部数据如长度、类型、协议及版本号等信息存放在 ftyp box 中；媒体数据也就是音视频元数据存放在 mdat box 中；最后，连续存放且杂乱无章的音视频数据还需要 moov box 中包含的各类媒体信息对其进行精确定位。图 4.25 为 MP4 文件中各 box 具体的层次结构。

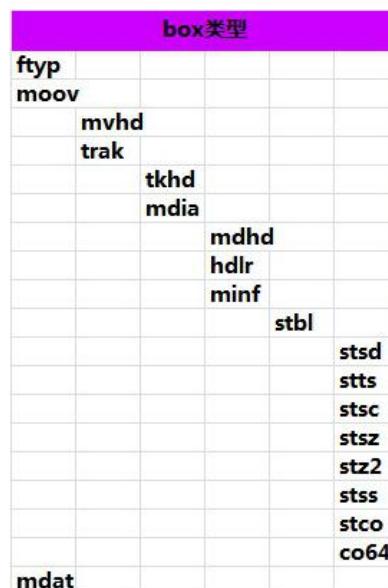


图 4.25 MP4 文件结构

在 MP4 数据流中，每开始一个新的 box，box 头都会表明其类型。因而对于视频的解封

装，客户端首先建立 MP4 数据结构并读取数据，在得到 box 类型后，使用相应结构体并结合从 box 读取的长度信息对 MP4 数据逐段复制、存放，直至数据读取完毕。因而，解封装过程相当于将 MP4 数据移植到了一个带有标签的框架中，以便于码流的提取。

3) 码流提取

之前提到，MP4 文件主要有三类 box，其中视频数据的层次结构，尤其是码流提取所需的数据存放顺序信息，由 moov box 中的元数据描述。码流提取要做的是根据图像块选择环境中获得的序号集，在 moov box 的相应通道 track 中寻找到帧索引，进而从 madt box 中梳理得到正确帧排序的视频数据。

从本节前述信息中可知，MP4 Box 封装好的多路 track 的 MP4 文件是 1+N 的多通道结构，第一路 track 中包含了 HEVC 头数据，主要是参数集信息（VPS, SPS, PPS），之后各路 track 含有按图 4.23 顺序排列的 tile 媒体元数据。因此，构建一个新的 FoV 区域码流，首先需要从第一路 track 中提取出参数集信息（VPS, SPS, PPS）作为新码流的头信息，再从其余 track 中得到码流在 mdat box 中的帧偏移位置和数据大小，最后依此从 mdat 裸数据池中进行抽取。

FoV 区域 tile 码流提取的具体算法流程如表 4.3 所示。

表 4.3 FoV 区域对应 tile 码流提取算法

输入：MP4 文件、tile 序号集

流程：

首先从第一路 track 中提取 VPS, SPS, PPS 等 HEVC 头信息

For (当前解码单元 (GOP) 的每一帧)

// GOP 表明了帧区间以及 I 帧间隔

For (每个所需的 tile)

1. 确定当前帧所需 tile 所处的数据通道

2. 确定当前帧数据属于哪个 chunk

3. 获得当前帧 tile 数据大小

4. 根据数据索引和大小从 mdat box 中提取出 tile 数据

最后，各帧数据拼接得到基于 FoV 的新数据流

码流提取以多帧组成的图像组 (GOP) 为单元进行。在之前提到过，编码模块设置了 IDR 帧间隔为 10 帧，因而传输的最小单元为 10 帧。在码流提取中，同样设定单个 GOP 的帧数为 10 帧，这样一个码流便可作为一个独立解码单元。要注意的是，由于数据的封装、传输，码流提取均基于 GOP，因而单个 GOP 将共用一个 tile 序号集进行码流提取以及服务器通道删除的操作。原型系统中初步设定以 I 帧状态提取的序号集作为整个 GOP 的 FoV 状态信息。

终端在抽取完 VPS, SPS, PPS 信息后，需要提取在一个独立解码单元中的 tile 数据。表 4.3 中算法建立了一个二次循环，以获取每个 tile 的同一帧在 mdat box 中的偏移位置以及数据大小。MP4 结构中存放这类元数据信息的具体 box 为 stbl，而真实的数据存在放在 mdat 中，stbl 与 mdat 之间有对应关系。

如图 4.25 所示，在 stbl 中有六个关键的子 box，其中，stts 存放了总帧数信息；stsz 存放了每个 Sample (也就是帧) 大小；stsc 是 Sample to chunk 的映射表，可以得到帧序号和 chunk 序号的映射关系，如图 4.26 所示，也就是说一个 chunk 中会包含多帧，因而要定位帧位置首先要找到包含其序号的 chunk；stco 是 chunk 位置偏移表；最后，stss 告诉了哪些帧是关键帧。

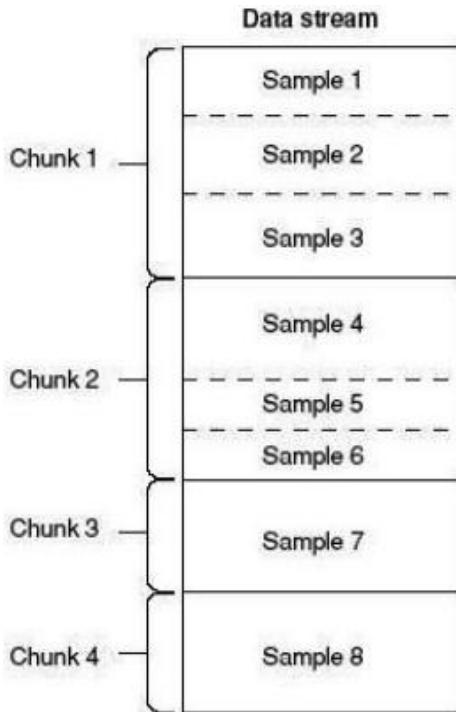


图 4.26 chunk 与 sample 的映射关系

因此结合表 4.3 算法, 如果要从 MP4 的某一路 track 中提取出第 M 帧的数据, 其过程如下:

1. 根据 Sample to chunk 的映射表, 找到第 M 帧对应的第 P 个 chunk;
2. chunk 的序号为 P, 根据 chunk 位置偏移表找到该 chunk 在文件中的偏移位置 a;
3. 获取帧序号 M 在第 P 个 chunk 中的位置, 并根据 stsz 表获知在 P chunk 中 M 帧之前的所有帧大小, 获取第 M 帧在第 P 个 chunk 中的偏移位置 b;
4. 由 2, 3 可知第 M 帧在 MP4 文件中的偏移位置 a+b;
5. 由 stsz 表得到第 M 帧的大小 c;
6. 根据 4 和 5 中得到的偏移位置 a+b 和帧大小 c, 在 mdat 中抽取出第 M 帧的数据。

由此, 可以在二次循环中抽取每一帧对应的 tile 数据, 直至独立解码单元的最后一帧, 便获得了整个 FoV 码流的数据。各帧数据进行简单拼接, 便可构建一个新的 FoV 区域码流。

4) 解码与裁剪

在获取到一个独立解码单元的 FoV 码流后, 系统可采用视频处理工具 ffmpeg 对当前时刻的码流进行解码操作, 获得 YUV 图像序列。

由于基于 tile 传输的特性, 解码后得到的 YUV 视频分辨率并不等同于 FoV 分辨率, 还需对边缘进行裁剪。其做法是对独立解码单元中的每一帧, 根据 FoV 视角信息文件中给定的当前帧位置坐标, 抽取完全对应于 FoV 大小的画面部分。最后将这一个独立解码单元的提取帧按照时间顺序拼接起来, 便组成了这一时间段内的 FoV 图像序列。根据本章前述步骤, 系统经长时间运行后会输出数个 YUV 子序列, 对于 YUV 数据, 将各段子数据按序首尾相接便可得到一个完整的基于用户视角播放的 YUV 视频。

通过上述多个步骤, 本节构建了如图 4.27 所示的整体动态传输流程, 形成了一个简单的基于 tile 的端到端系统。

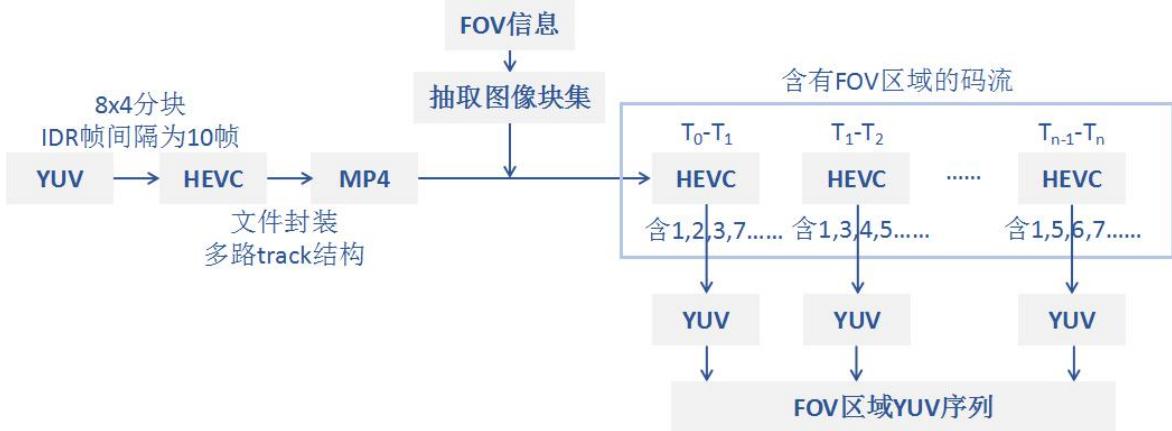


图 4.27 系统动态解码流程

4.3.2 GPAC: 使用 tile 的 VR/360 视频传输

GPAC 实际上也给出过全景视频的传输实例。与大部分方案类似，GPAC 大体从以下几个方面降低传输带宽，但具体做法会有所区别：

- 视频压缩
 - 经投影后的 2D 视频压缩
 - 封装层面的压缩
- 自适应传输
 - 基于视角的传输，视角外提供低质量图像
 - 对视角移动的快速反应

在 ERP 映射格式下，GPAC 可以对 ERP 视频进行特殊封装，如下图所示。



图 4.28 (a) ERP 投影; (b) ERP 格式特殊封装

在上一节中提到过，GPAC MP4Box 可以实现 tile-track 一对一的封装，实际上，GPAC 也提供了多 tile 单路封装的方法。

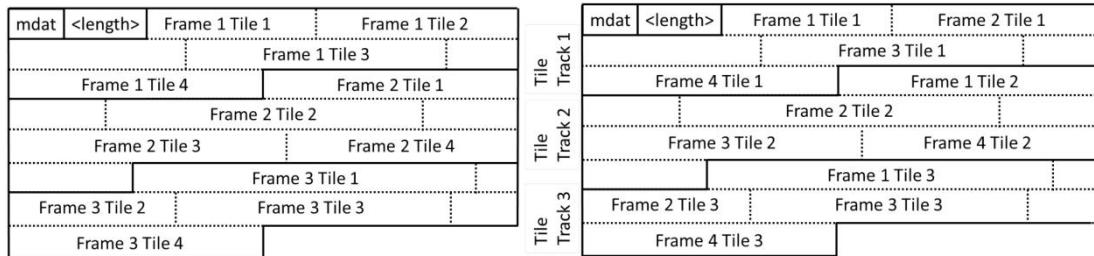


图 4.29 单路视频 track 和多路视频 track 文件的存储比较

如图 4.29 所示为单路视频 track 和多路视频 track 文件的存储比较。从图中可看到，

单路 track 的 MP4 文件的 mdat 存储数据的顺序是以帧为大结构，在帧内按照 tile 结构顺序存储；多路 track 结构在上节中介绍过，其保证了 tile 数据的连续存储，有利于动态抽取，使得抽取某路 track 的 HEVC 更加便捷。为契合 OMAF 等标准思想，一般采用多路 track 方法封装。

GPAC 采用之前章节提到的 DASH 流协议进行传输，由于全景视频的特殊性、GPAC 的灵活性，可以根据质量等因素分成双流/多流传输，如下图所示。



图 4.30 双流传输

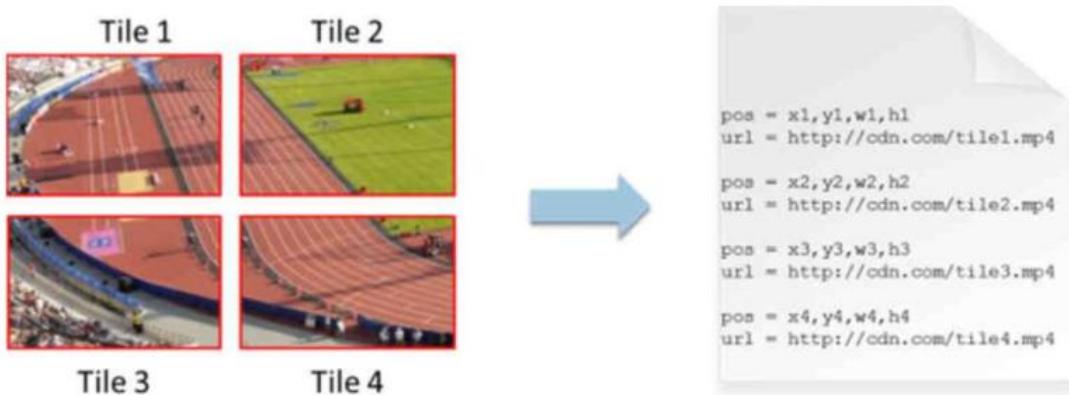


图 4.31 SRD 文件

而在 Dash 传输中，存在一项重要的说明文件，名为 SRD，如图 4.31。该文件主要描述了数据间空间关系，以下为一个 SRD 描述例子：

```

MP4Box -dash 1000 [other dash params]
source.mp4: desc_as = <SupplementalProperty
schemeIdUri = \ “urn: mpeg: dash: srd: 2014 \” value = \ “0, 0, 1, 1, 1, 2, 2 \” />

```

此句式表示 source.mp4 文件位于 X = 0, Y = 1, 宽度为 1, 高度为 1, 大小为 2×2 的 tile 网格中。独立视频此信息一般需要人为确定，但如果源文件包含 HEVC tile track，则会自动插入 SRD 信息。我们也可以从 GPAC 中带有 GUI 的播放器 MP4 Client 中观察到不同的 tile 质量和统计数据：

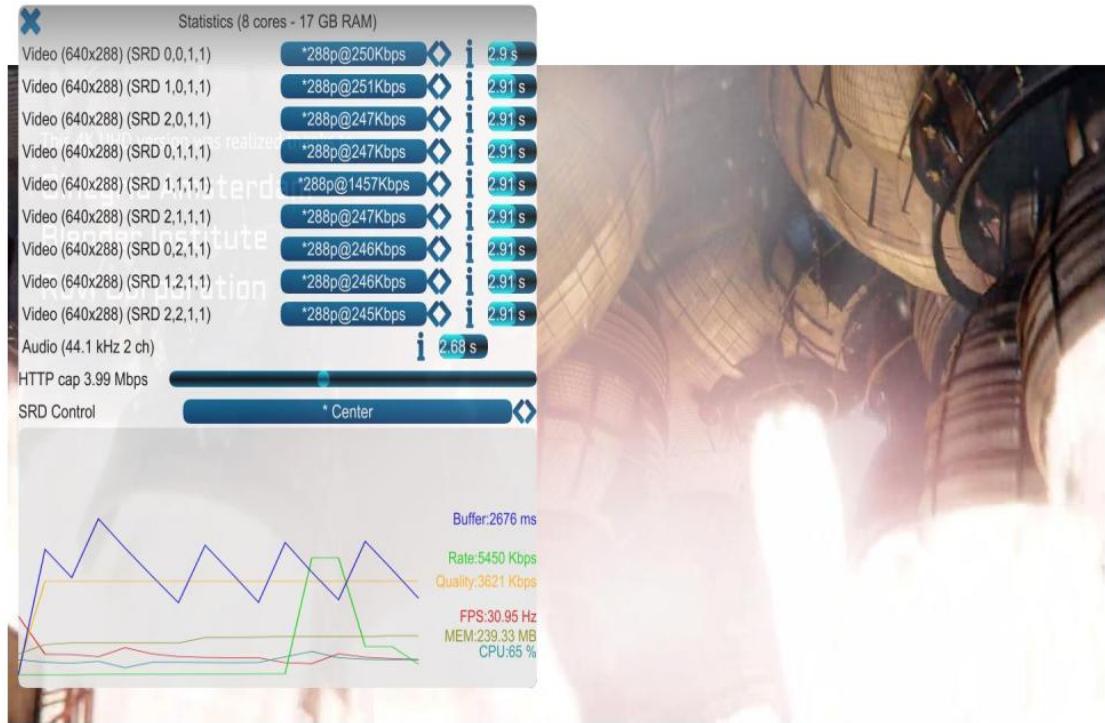


图 4.32 带有 GUI 的播放器 MP4 Client

结合上述内容，GPAC 基于 HEVC 标准的自适应传输流程如下所示：

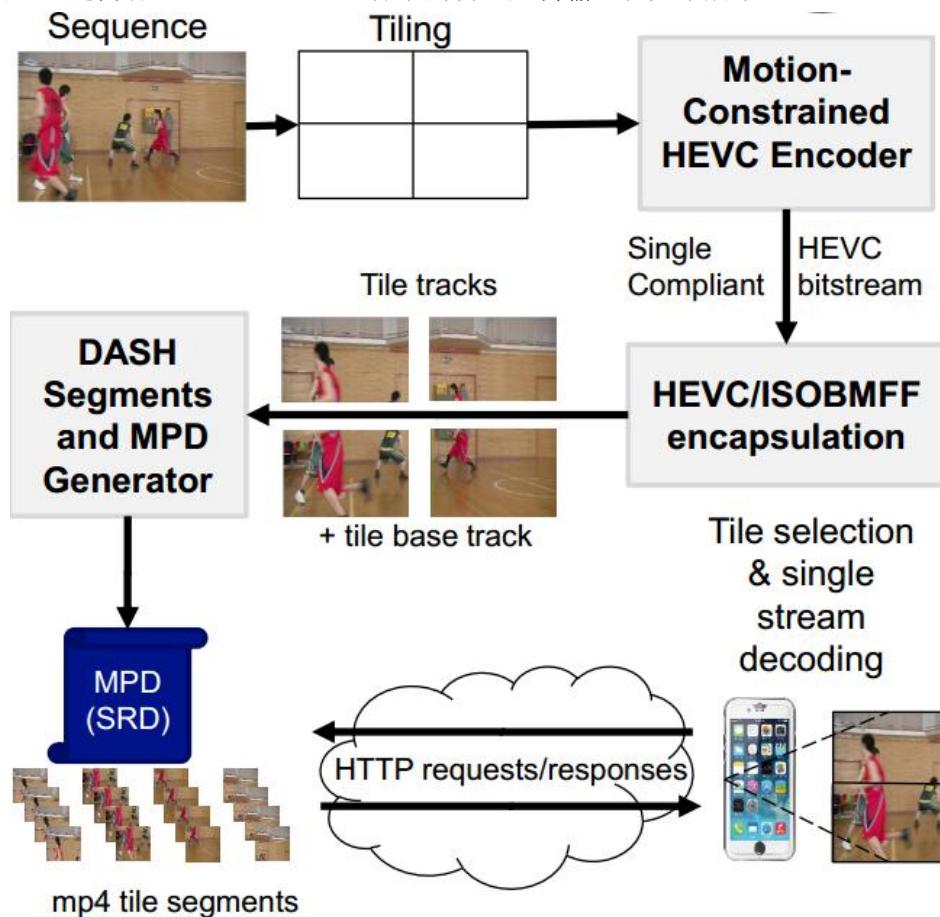


图 4.33 GPAC 的自适应传输流程

4.3.3 360 视频流媒体传输系统——Two-Tier Streaming (TTS)

作为图像/视频编码领域的顶级会议之一,第33届图像编码研讨会(PCS,Picture Coding Symposium)于2018年6月24号至6月27号在加州旧金山召开。会议旨在为视觉压缩领域提供一些突破性的先进技术以及提供高水平的学术报告。在会上,纽约大学工学院教授Yao Wang做了关于最新的360视频流媒体传输系统TTS的主题报告,介绍了当前TTS的测试情况以及后续研究计划。

Two-Tier Streaming 概要

对于基于DASH的2D视频流媒体点播,视频首先从时间上被分为多个子片段,并以多个分辨率版本存储于服务器。客户端则根据网络吞吐量和缓冲区长度请求所需数据。其中,客户端存在一个预提取的操作,其可以检测网络变化并应对突发情况,是较为关键的一个环节。

对于360视频,由于用户只能观看到FoV中的场景,因而目前的各类360视频流媒体解决方案大多通过传输当前和预测FoV对应画面的形式,而不再传输完整的全景内容,以减少带宽浪费,提高传输效率。然而,目前的FoV预测仍会引起预测偏差和卡顿的问题。

由Yao Wang教授报告,华为和NYU WIRELESS团队共同完成的Two-Tier 360V Streaming系统则结合了预提取与FoV预测过程,对360视频传输做了以下改进:

- 双层编码:
 - 基础层(BT)数据: 包含低质量的完整360场景
 - 增强层(ET)数据: 包含多视角的多种比特率场景
- 双层传输:
 - 利用长预提取缓冲区(10–20s)下载BT数据
 - 利用短预提取缓冲区(10–20s)下载基于FoV预测ET数据
- 双层渲染:
 - 如ET数据与实际FoV匹配,则对FoV渲染高质量视频
 - 否则,利用BT数据对FoV渲染低质量视频

基础层主要针对网络与视角的动态特性提供良好的鲁棒性。

- 数据区分与编码:
 - 未重叠区域编码: 无存储冗余, 低编码效率
 - 重叠区域编码: 高存储冗余, 高编码效率

BT与ET数据间的分层/非分层编码方式实际上是寻求编码效率与复杂度平衡点的问题。

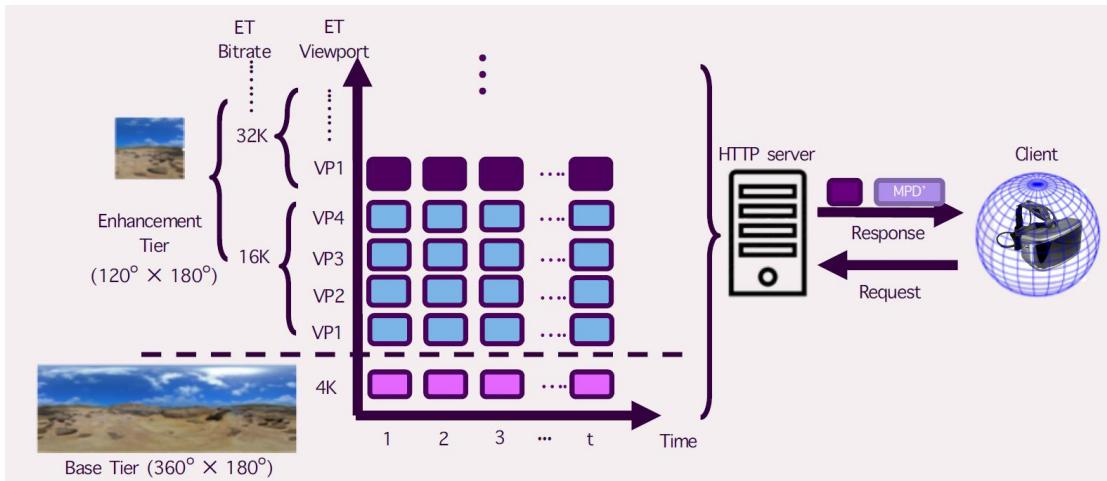


图 4.34 TTS 系统

系统关键技术

作为一个标准的流媒体传输系统，TTS 主要包含以下关键问题或技术：

- 速率分配：如何设置两个层在传输中的速率？
- 传输决策：两层缓冲区各为多长？下载/提取哪部分数据？
- 多目标优化：基于网络与 FoV 动态特性的视频质量、连续性、响应性

在 TTS 中，渲染后的视频总质量取决于 BT/ET 层质量以及 FoV 预测吻合率 α 、数据块传输速率 γ 。在前述参数确定的情况下，视频质量仅为各层比特率的函数，根据导数条件也就是图像质量最优条件，可以得到两种比特率的关系，便达到了速率分配优化的效果。而在比特率确定， α 、 γ 待定的条件下，该团队通过一系列控制变量的测试得出了如下结论：为获得最优视频质量， α 与 γ 的乘积应为最大。

为获取初步结果，研究团队采用了传输完整 360 度内容的 Benchmark System 1 (BS1) 和仅传输经线性预测的 FoV 对应内容的 Benchmark System 2 (BS2) 作为对比。在相同的测试条件 (5G WiGig 网络，多类型场景等) 下，TTS 相比于 BS1 具有更高的视频渲染率 (VRR)，不同网络情况下可以提高 275%-470% 不等，同时卡顿率相差无几；相比于 BS2，其具有同样级别的 VRR，而卡顿率可以下降 2%-21% 不等。此外，随着传输环境的恶化，上节提到的 TTS 最优 $\alpha\gamma$ 值会降低，系统将分配给 BT 层更多的带宽。速率分配和缓冲区优化均可以提升用户体验质量 (QoE)。

TTS 中设置了 FoV 校正步骤，即对于即将播放的画面进行二次预测，以弥补图像缺失部分，由此提升的效果取决于校正范围和校正预留时间。对于流媒体点播而言，每个数据块可以包含经预测得到的未来视频片段，同时应尽可能地提前抽取出播放部分。

然而现有的 FoV 预测方法还难以实现长间隔 (数秒) 的准确预测效果，主要有以下几种：

- 仅利用用户过去的 FoV 轨迹
- 同时利用视频内容和过去的轨迹
- 采用目标用户的过去轨迹以及其他用户的已知完整轨迹
- 嵌入机器学习

测试结果显示，利用多用户轨迹完成的预测比单用户轨迹预测的效果更好，且随预测间

隔的增大，提升的准确度越高，最高可达 8%左右。

360 视频流媒体传输中的另一个关键问题是传输决策，合理的传输方案可以有效减缓网络负担，同时保证良好的 QoE。对于 TTS，基于数据块的传输决策主要体现在：

- 当一个数据块到达时
 - 下一数据块的类型：BT/ET？
 - BT/ET 块的比特率/质量水平及其对应的 FoV
- 过程简化
 - FoV 预测的独立性
 - 传输决策仅为数据块提取与速率选择提供服务

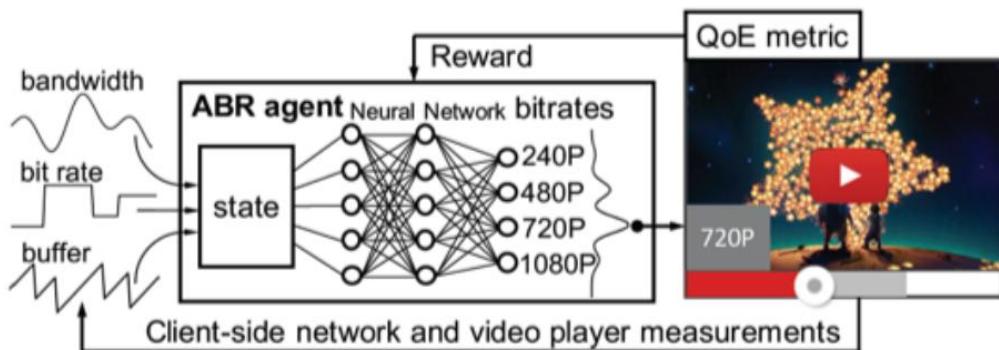


图 4.35 基于强化学习的传输决策

在 TTS 中，数据块的抽取问题被看作是一个强化学习的过程。该过程主要考虑到速率与网络性能的平衡以及各决策对于未来的影响程度，而各类状态如缓冲区大小、吞吐量、视频质量等可以看作不同的变量进行优化。在二维视频传输中，已有深度强化学习方法采用了基于 QoE 指标的神经网络模型，并构建了校正网络（Critic Network）和动作网络（Actor Network）以共同完成传输决策。实际上，类似的方法也可移植到 TTS 上，但这种移植应考虑到 TTS 的额外状态变量如 BT/ET 的缓冲区和比特率，以及更复杂的反馈机制。

小结

360 视频的诞生为视频编码/传输领域带来了许多新的挑战。在编码与传输紧密结合的基础上，华为和 NYU WIRELESS 的团队共同搭建了 360 视频流媒体传输系统 TTS，其双层处理的概念便于码率分配、质量优化、传输决策等后续过程，同时在视频渲染率、卡顿率等指标上有明显的提升，对于网络和 FoV 的动态特点具有良好的鲁棒性。

360 视频流媒体传输三种应用场景的约束各不相同，TTS 主要针对流媒体点播的形式进行了改善，当应用至交互式与直播场景时，还应考虑到随机性、实时性、准确性、多路性方面更严格的要求。因而，360 视频流媒体距离多方位、理想化的实现还有一段路要走，但也是有可能实现的。

4.3.4 8K VR 视频系统

2018 年 7 月 5 日，NTT DOCOMO 宣布成功开发出全球首个基于 5G 网络的 8K 超高清 VR 实时视频流式传输及观看系统，可将其用于对赛事、音乐会等的 360 度全景 8K VR 视频实时直播。

NTT DOCOMO 称，在终端侧，用户戴上头显设备后可享受流畅逼真的 VR 体验，以观看现

场音乐和体育赛事。

整个系统如下图所示：

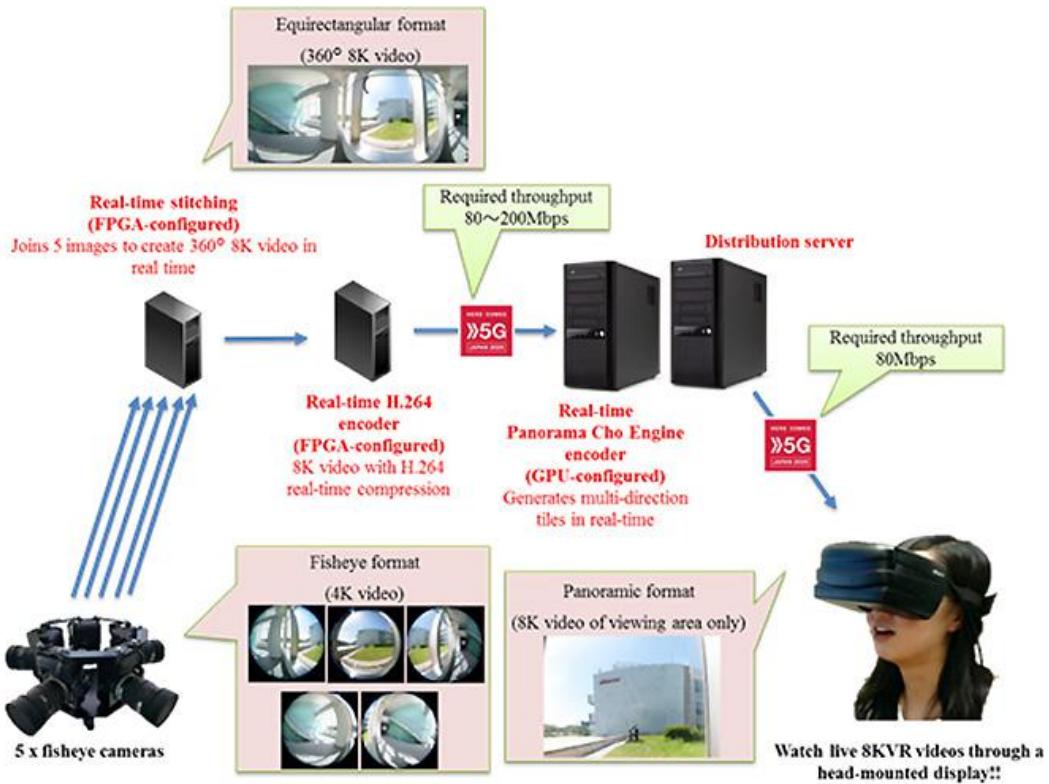


图 4.36 8K 超高清 VR 实时视频流式传输及观看系统

摄像机一共有 5 个 4K 镜头，用以拍摄出 360 度全景 4K 视频。其中，用 $4K \times 4K$ 方式拼接成 8K 视频，最终传输 8K VR 直播视频流，这个过程不仅需要消耗大量带宽，同时对于实时处理/计算的时延有着非常苛刻的需求。

为降低时延，NTT DOCOMO 并未采用软件方式来执行上述计算，而是用 FPGA 硬件+高密度 30 帧每秒算法来实现；然后再用实时 H.264 编码器（同样是用 FPGA 硬件来实现）对 360 度全景 8K VR 视频进行压缩，至 80~200Mbps；再用 5G 网络将其从音乐会、赛事等大型活动的现场回传至媒体中心前端；再在前端用实时全景 Cho 引擎编码器把 360 度全景 8K VR 视频切割成多个空间方向的片段（以便用户在戴上头显设备后观看 360 度视频时，随着头部的移动能看到对应的影像）；最后以 80Mbps 的码流速率通过 5G 网络传送至用户的 360 度全景 8K VR 头显设备。

NTT Docomo 计划在 Docomo 5G Open Lab Yotsuya 展出该系统。5G 的展示空间提供了高频率音频和视频的体验，能够为用户带来良好体验。如果未来在火车等交通工具内部署了 5G 设备，它便可以提供更好的乘车体验。有了这些部署以后，对消费者来说以后就不用担心错过任何精彩瞬间。

4.4 沉浸式流媒体优化技术

4.4.1 快速屏幕内容编码 (FSCC)

由于移动端和云应用技术的快速发展，屏幕内容视频 (Screen Content Videos, SCV) 越来越多地出现在人们视野中。而在许多应用场景中，SCV 的实时传输显得尤为重要。对此，JCTVC (Joint Collaborative Team on Video Coding) 工作组从 2014 年开始，在 HEVC 编码标准的基础上，就屏幕内容编码 (SCC) 进行标准扩展，并于 2016 年完成这项工作。

屏幕内容有许多形式，如文字、曲线、图案、人脸等等。近年来，SCC 又出现了许多新模式。帧内块复制模式（IBC，图 4.37）基于屏幕文字样式重复的特性，并采用了帧内快搜索和运动补偿方法来降低数据量。另一种画板编码模式（PLT，图 4.38）则结合颜色 RGB 表和对应的索引表，生成一个综合值来表示相应的内容块，这种模式主要是利用到屏幕颜色低质量或质量可区分的特性来进行压缩的。

in SCC, namely intra palette coding (CPC).
a CU (coding unit) will



图 4.37 IBC 模式



图 4.38 PLT 模式

类似的新型 SCC 编码模式已被证明相对于 HEVC，可以节省 50% 的比特率，是目前最有效的屏幕内容压缩方案。然而，这类方案目前存在的问题是高计算复杂度，主要是由编码中的块分割和模式选择环节引起，还需要进一步改善。

常规的编码器会逐级检测和比较每种编码模式的代价，决定最佳的分割模式。2015 年，F. Duanmu 等人则提出了一种基于预训练神经网络的快速 CU 分割模式决策算法，网络的输入特征包括颜色数量、梯度峰态、CU 差异、子 CU 差异等，可以达到快速确定 CU 是否继续分割的效果，该算法可以降低帧内编码 37% 的复杂度。在此基础上，该团队在 2016 年对此方法进行了改进，新增了区分自然图像块（NIB）和屏幕内容块（SCB）的分类器和区分定向/非定向的帧内块分类器，最终形成了一种快速屏幕内容编码（FSCC）法，可以降低最高 52% 帧内编码的复杂度。

快速屏幕内容转码

2016 年 F. Duanmu 等人的团队研究了一种快速 HEVC-SCC 转码方式，该方法利用 CU 特征和 HEVC 解码端信息训练块分类器，重复利用 HEVC 编码深度可以推测 SCC 编码深度，最终可以在 BD-Rate 少量损失的情况下，降低 48% 帧内转码复杂度。

实际上，目前的一些传统设备尚未支持 SCC 比特流的解码，市面上也需要降低转码复杂度且保证效率的 SCC-HEVC 转码算法，以兼容新兴的屏幕内容应用。

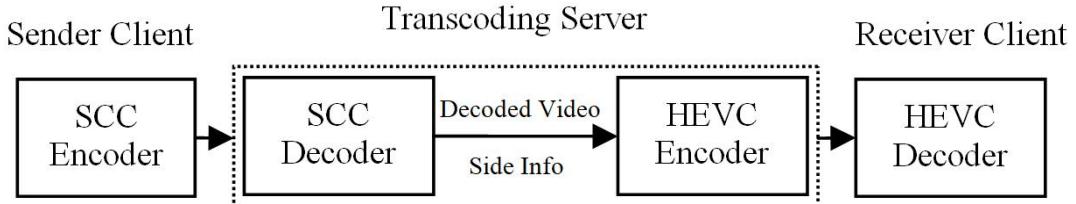


图 4.39 快速 SCC-HEVC 转码

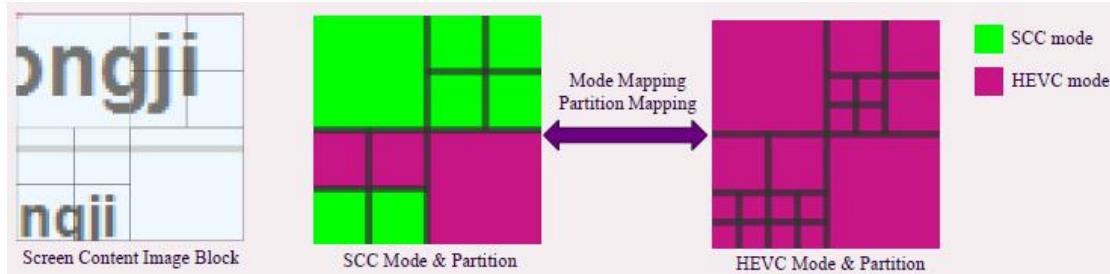


图 4.40 编码块分类与分割

F. Duanmu 等人建立的 SCC-HEVC 转码系统将 SCC 流的帧内模式、帧间信息（运动向量、参考图像集、参考帧索引等）直接给予 HEVC 编码器使用。PLT 模式的转码方式根据解码的索引映射表推得块结构及其方向性，并触发帧内模式的快速选择。而 IBC 模式经解码后会产生块向量（BV），以确定与先前编码区（或当前帧）相匹配的区域。如当前块与匹配区域的帧内模式相同，则沿用该模式；否则直接对当前块（CU 层面）进行分割。如当前块与匹配区域的帧间模式相同，则根据 BV 和匹配区域的 MV 共同得到最终的 MV。经测试，该系统在全帧内和低延迟模式下分别可以实现 2 倍和 5 倍和转码速度。

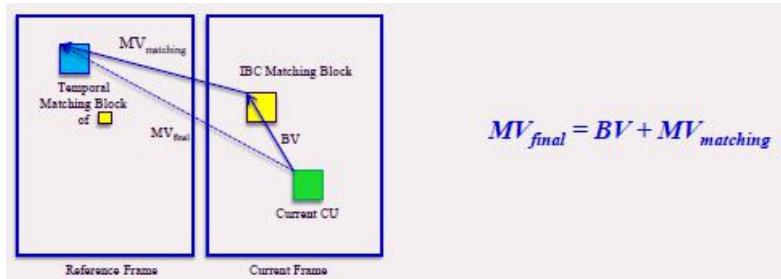


图 4.41 帧间 IBC 模式转码

此外，SC 转码也可以采用如图 4.42 所示的单输入多输出（SIMO）的网络。转码器可以将单一高比特率的 SCC 流转为多个质量递减的 HEVC 流，这一过程支持并行操作。分级转码的原因在于 NIB 对于量化参数 QP 的敏感程度远高于 SCB，因而 SCB 编码深度的不匹配不会引入过多的视觉质量和编码效率损失。SIMO 网络可以最小化核心带宽消耗以及边缘缓冲区存储，同时边缘计算复杂度和系统处理延迟都较低。

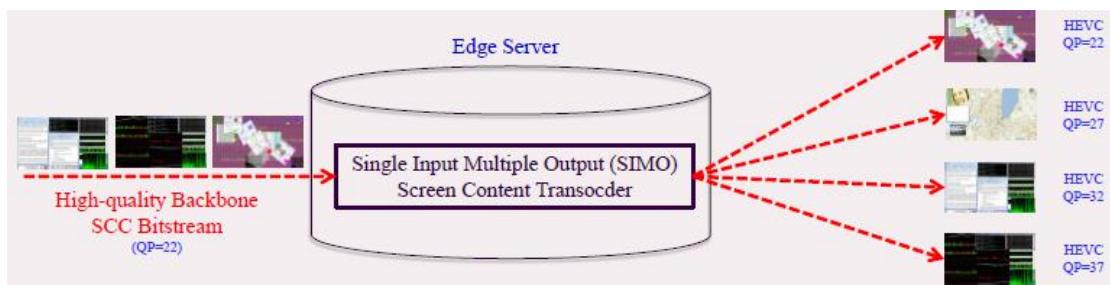


图 4.42 SIMO 转码网络

4.4.2 VR+5G

自2014年三星电子开发出首个基于5G核心技术的移动传输网络开始，5G就成为了科技业界内的热点。作为一项颠覆性技术，5G将极大地改变人们的生活方式，并在VR/AR、自动驾驶、智能假造等场景中有广泛应用。

根据华为发布的《5G时代十大应用场景的白皮书》，云VR/AR被列为5G时代最值得期待的应用场景之一。那么，5G技术到底能为VR/AR提供哪些支持？5G时代，AR/VR体验又将拓展出哪些值得期待的应用场景？作为4G网络的升级版，5G何以成为VR/AR市场的引爆点？



图4.43 5G网络概念（1）

5G推动VR发展的关键——高速传输

目前，智能手机终端的VR/AR应用多数是基于独立的APP运行。就观看VR视频为例，一段几秒钟的高清全景视频便可达到几十兆甚至几百兆。在主流的4G网络的传输速度下，用户是难以流畅观看VR视频的。

而对于AR体验来说，虽然可以依靠离线的识别处理机制来呈现虚实结合的体验，但当识别的景象发生连续大量的动态变化时，单单依靠终端便难以负荷庞大的计算量。

为此，华为VR OpenLab联合视博云等合作伙伴在2018年2月举办的西班牙MWC展览会上，发布了最新的VR解决方案——Cloud VR，即将VR运行能力由终端向云端进行转移，以此来推动VR/AR应用在智能手机端的普及。

然而，这种解决方案的实现所依托的仍然是高效的传输网络——5G。



图 4.44 5G 网络概念（2）

所谓 5G，指的是第五代移动通信网络，其主要目标是打破当前无线网络中范围限制的壁垒，真正让用户能够拥有实时联网、随处可用的体验。

作为 4G 网络的真正升级版，5G 最大的特点便是采用特高频进行通信。根据国家工信部的规定，我国的 5G 初始中频频段为 3.3–3.6GHz 和 4.8–5GHz 两个频段，同时，24.75–27.5GHz、37–42.5GHz 高频频段也正在征集意见。而当下主流的 4G LTE 所采用的却多是 0.3–3GHz 频段。

名称	符号	频率	波段	波长	主要用途
甚低频	VLF	3-30KHz	超长波	1000Km-100Km	海岸潜艇通信；远距离通信；超远距离导航
低频	LF	30-300KHz	长波	10Km-1Km	越洋通信；中距离通信；地下岩层通信；远距离导航
中频	MF	0.3-3MHz	中波	1Km-100m	船用通信；业余无线电通信； 移动通信 ；中距离导航
高频	HF	3-30MHz	短波	100m-10m	远距离短波通信；国际定点通信； 移动通信
甚高频	VHF	30-300MHz	米波	10m-1m	电离层散射；流星余迹通信；人造电离层通信；对空间飞行体通信； 移动通信
超高频	UHF	0.3-3GHz	分米波	1m-0.1m	小容量微波中继通信；对流层散射通信；中容量微波通信； 移动通信
特高频	SHF	3-30GHz	厘米波	10cm-1cm	大容量微波中继通信；大容量微波中继通信； 数字通信 ；卫星通信；国际海事卫星通信
极高频	EHF	30-300GHz	毫米波	10mm-1mm	再入大气层时的通信；波导通信

图 4.45 频段说明

也就是说，5G 所采用的频率是远高于 4G 网络的。而频率越高，频段就越宽。频段加宽，就可以使单位传输量得到大幅度提升，进而带来超高速的传输速率（可以将频段理解为车道，车道加宽，速度自然提升）。

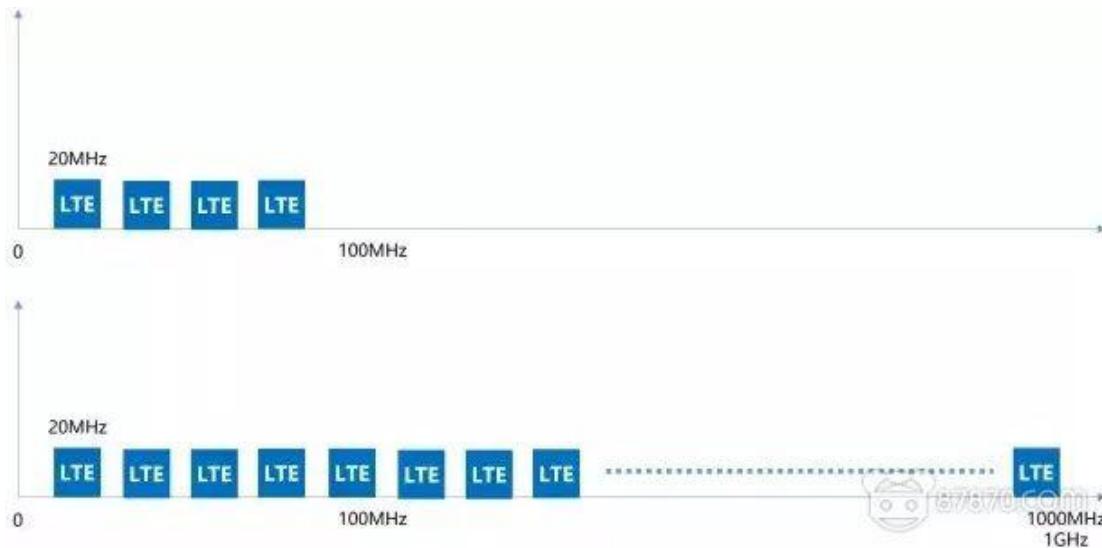


图 4.46 频段推进

从理论上来讲，5G 网络的最高传输速度可高达每秒数十 Gb。目前，三星电子所研发的 5G 网络已成功实现在 28GHZ 的频段下达到速度 1Gb、范围 2Km 以内的数据传输（当前 4G 服务的传输速度约为 0.06G）。

这个概念就意味着，在 5G 时代，一部超高清的电影可在 1 秒之内下载完成。同样，一段超高清的 VR 全景视频也可以实现实时的流畅播放。

5G 优化 VR 体验的核心——微基站



图 4.47 5G 网络概念 (3)

但是，仅仅“快”仍无法解决 VR/AR 体验在移动终端中的延迟问题。事实上，5G 网络还在其整体设计上采用了不同于 4G 网络的基站布局和处理机制，以此来缩短传统 VR 体验中的延迟时间。

对传输网络来说，所采用的频率越高，传播过程中的衰减也越大，这就导致了 5G 网络覆盖能力的减弱。所以，同 4G 网相比，5G 所需要的基站数量将更加庞大。但同时，覆盖范围的缩小也减轻了基站所承载的传输压力。因此，相比于 4G 网络建造的宏基站，5G 网络所

采用的基站更多的是微型基站。



图 4.48 网络基站

在未来的 5G 时代，微基站随处可见，它将遍布在各个生活和工作场所。这也意味着，5G 网络的基站将距离用户更近。它可能就藏在用户的办公楼里、用户休息的咖啡厅里、甚至离用户的家仅几步之遥。

基于微基站，5G 采用移动边缘计算机制，即将处理逻辑下沉到网络的边缘，也就是更靠近用户的基站上。一旦用户发出请求，数据便可以在极短的时间内传输到基站，而基站也可以更快速地给用户以反馈。

正是基于这种高效的传输机制，5G 网络才能够让 VR/AR 应用在移动终端的时延极大地缩短。根据 IMT-2020 制定的指导方针，5G 将提供 1 毫秒的 OTA 往返延迟。实际上，当延迟小于 10 毫秒时，人类就基本无法察觉到画面的延迟。因此，5G 的到来将会彻底消除 VR 使用中由时延所带来的眩晕感，从而真正提升移动终端的虚拟体验。



图 4.49 5G 网络概念（4）

北京邮电大学网络与交换技术国家重点实验网络服务基础研究中心副主任齐秀全在新华网的采访中曾说：“5G 能够给我们带来很多不同的业务体验形式和技术上的保障，从不同角度支撑高宽带、低延时和计算密集型业务的开展。到 2020 年 5G 正式商用之际，AR/VR 体验将成为市场主流。”

5G 时代，更多的 VR 应用场景将成为现实



图 4.50 通过 VR 观看各类直播

正如前文所言，4G 网络仅能够满足部分 VR/AR 应用，但 5G 时代的到来不仅增强了现有的虚拟体验，还将拓展出全新的应用场景，真正使 VR/AR 发挥其在移动终端的优势，解决用户生活中的痛点。

比如，目前发展缓慢的 VR 直播。囿于 4G 网络环境的带宽限制，用户无法仅靠移动终端来实现体育赛事和演唱会等大型场景的现场直播，即使采用专用级的 VR 全景摄影机来进行视频采集，用户终端的观看体验也仍然欠佳。但随着 5G 时代的来临，高清 VR 视频的上传和在线播放的流畅性都将在几秒之内完成。

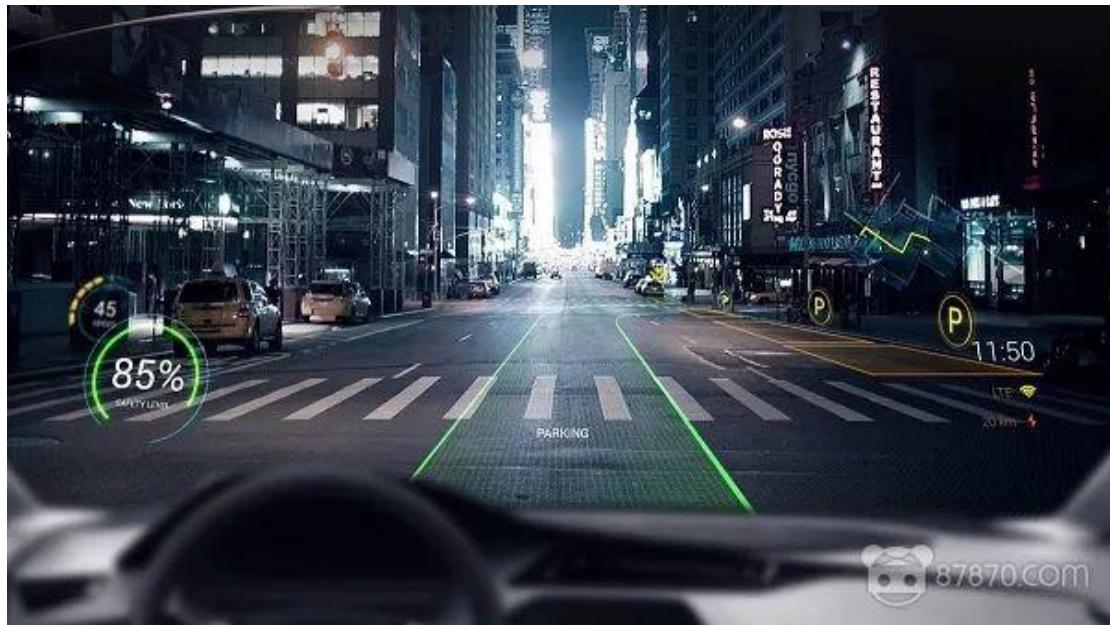


图 4.51 基于 AR 的车载导航

同时，5G 网络还可以使基于 AR 的车载导航成为现实。将导航地图和实时路况等信息投射在驾驶员眼前的挡风玻璃上，使驾驶员在搜索路线的同时也能够对行驶道路的状况进行把控，从而既提升了行驶的安全性，也节省了驾驶时间。

此外，随着 5G 的部署，一些对实时性要求较高的应用，诸如远程手术、虚拟课堂培训和即时 VR 内容创作等，也都将得到普及。正如乔秀全教授所说，AR/VR 在 5G 时代的发展将为我们开启一个全新的时代。

4.4.3 混合质量传输方案

实际上，5G 的到来仍然需要一段时间。由于诸如网络等多条件的限制，目前全景视频的传输会采用到一种混合质量传输方案（4.1 节和 4.3 节中均有提到），即为全景图的不同部分提供不同的质量（或分辨率）等级。该方案中有一个重要的模块，在这里被称为 tile 选择器。

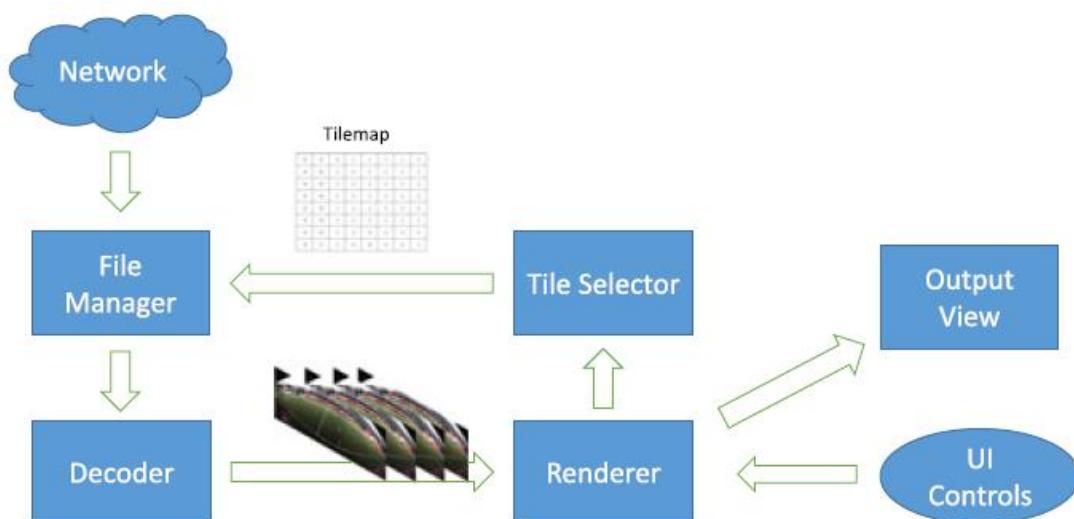


图 4.52 基于 tile 的客户端架构，其中包含了 tile 选择器

tile 选择器：每显示一个帧后，渲染器将提取当前视图的全景位置，并传送至 tile 选

择器。此信息对于选择下一组 tile 至关重要。因而，此模块需要实时执行以向用户提供良好的交互式体验，同时将带宽消耗保持在较低水平，是多视频实时解码的客户端中一项具有挑战性的任务。

tile 选择器负责确定所需的不同类型 tile 的质量（比特率），并根据用户的移动进行调整。设 $Q = \{q_0, q_1, \dots, q_{n-1}\}$ 是 n 个可用质量等级的集合，其中 q_0 表示最高质量，质量按递减顺序排列， T_i 是第 i 个 tile 的质量。该问题则可以写成等式 (4.4) 中的简单标记问题，如下所示：

$$T_i = q, q \in Q \quad (4.4)$$

有多种方法可以执行该标记过程，所选方法最终也将影响所消耗的带宽和用户体验。标记过程一般采用二元 tile 映射，其包含当前用于生成虚拟视图的 tile 信息。当视图需要全景图上的第 i 个 tile 时，二元 tile 映射具有 $B_i = 1$ 。基于二元映射的基础上，接下来将简要概述一些 tile 质量选择方法。前三个方法主要对预定义或可配置的高/低质量 tile 做出二元决策。最后一种方法允许根据 tile 的重要性逐渐（多级）降低质量。

1) 二元法

为满足视频最基本的完整性要求，选择器最直接也是最易实现的是二元 tile 质量选择方法，即为了保证全景视频观看的沉浸感，选择器为用户正要观看到的部分提供高质量图像，同时为防止出现图像缺失情况，对其他区域覆盖低质量作为运动保护，如图 4.53 所示。

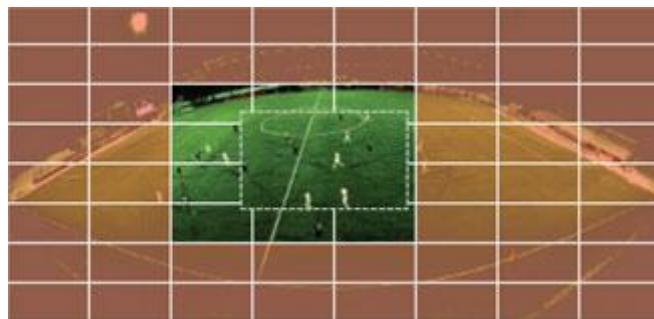


图 4.53 二元法示意图

该方法等价为如下关系式：

$$T_i = \begin{cases} q_h, & B_i = 1 \\ q_l, & B_i \neq 1 \end{cases} \quad (4.5)$$

在这种情况下应满足 $l > h$ ，但具体的质量仍可以根据实际环境进行调整。

2) 缩放法

在关于 tile 的研究中，另一种常用的方法是发送基础的低质量缩略视频，同时仅提供所需的高质量 tile，如图 4.54 所示。要创建缩略视频，就需要先缩小再存储源视频。在虚拟视图生成过程中，使用高质量 tile 填充视野内及边缘像素。对于无对应高质量数据的像素，缩放视频被放大后使用，这可等效为是低质量的 tile。

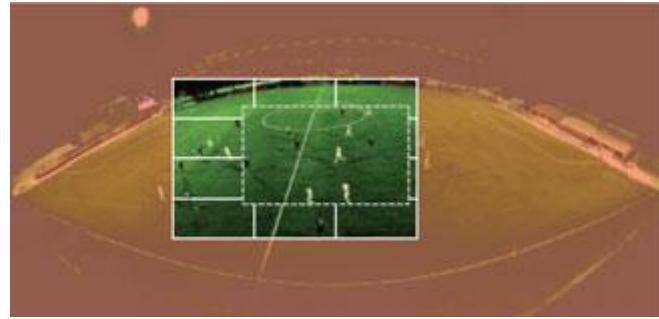


图 4.54 缩放法示意图

3) FoV 预测法

当用户观看全景视频向一个确定方向移动时，由于 tile 质量在边界处改变。有可能由一些低质量的 tile 生成视图。为了降低发生这种情况的可能性，可以对用户视角 FoV 进行预测，得到高可能性的未来运动方向，后提高该方向上邻近 FoV 区域的 tile 质量，以作为一种更具有针对性，更符合全景视频观看特性的运动保护措施，如图 4.55 所示。这类似于二元法，但它根据预测扩大了高质量区域。所以，对未来帧视角移动进行预测是存在收益的。

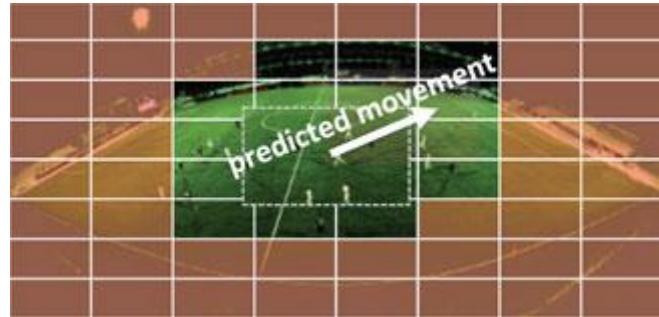


图 4.55 FoV 预测法示意图

FoV 预测模型

就 FoV 预测而言，有多种模型可供选择。最经典，同时与现有技术具有一致性的是自回归移动平均（ARMA）预测模型。对于这种方法，令 θ_t 为位置， $\delta\theta_t$ 为时刻 t 的视角移动速度。

则 $\delta\theta_t$ 可以由下式估计得：

$$\delta\theta_t = \alpha\delta\theta_{t-1} + (1-\alpha)(\theta_t - \theta_{t-1}) \quad (4.6)$$

进一步， $t+f$ 时刻的未来位置 $\hat{\theta}_{t+f}$ 可以估计得：

$$\hat{\theta}_{t+f} = \theta_t + f\delta\theta_t \quad (4.7)$$

其中 f 是预测间隔的帧数。这种预测结果可以立即用于构建未来的二元 tile 映射，并且该映射可以用于其他的质量选择方法。

然而，神经网络、显著性检测等特征提取、数据预测技术在近年来快速发展，并已被证明具有比传统回归模型更好的指示、预测效果，在图像/视频处理领域得到了广泛应用。类似的方法也同样适用于 FoV 预测。

采用神经网络进行预测时，首先应确定训练的数据类型。由于欧拉角的自相关特性和其

意义的简洁性，许多 FoV 预测相关的研究中常使用这类角度进行数据训练。

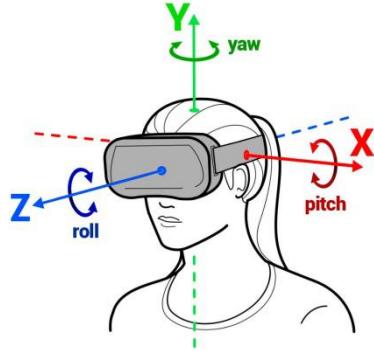


图 4.56 欧拉角头部运动模型

以图 4.56 中的 pitch 和 yaw 为例，令 X_i 、 Y_i 分别为第 i 帧对应的 pitch 角和 yaw 角，再 $X_{i_1:i_2}$ 、 $Y_{i_1:i_2}$ 令为第 i_1 帧至第 i_2 帧按序排列的角度集合。神经网络要做的是对于第 i 帧，在系统获得 $X_{i-I:i}$ 、 $Y_{i-I:i}$ 后，预测得到数据对 (X_{i+I_w}, Y_{i+I_w}) ，其中， I 表示预测窗口大小， I_w 表示当前帧与预测帧间隔的帧数。由于独立欧拉角自相关性远强于角度之间的相关性，一般应建立两个独立模型分别进行预测，因而输入输出角度之间的关系可以用以下关系式表达。其中， \hat{X}_{i+I_w} 、 \hat{Y}_{i+I_w} 为两种角度的预测值。

$$\begin{cases} \hat{X}_{i+I_w} = f_{I_w}(X_{i-10:i}) \\ \hat{Y}_{i+I_w} = f_{I_w}(Y_{i-10:i}) \end{cases} \quad (4.8)$$

这实际上是一个序列回归问题，而神经网络通常也适用于解决这类问题。然而，欧拉角存在连续性特点：其取值区间为 $(-180^\circ, 180^\circ]$ ，区间两端值从数值角度而言相差约 360 度，但其物理意义上的状态相差无几。如果直接采用欧拉角数据进行网络训练，数值意义上的巨大差别会引起较大的误差。为了使预测能兼容欧拉角的连续性特点，可将欧拉角转化为连续的坐标数据。一个可行的做法是将欧拉角映射至相应的单位圆坐标上，通过坐标对进行训练、预测。对于任意的一个欧拉角 θ_i ，其在单位圆上的坐标为 $(P_{i,1}, P_{i,2})$ ，两者映射关系如式 4.9 所示：

$$\begin{cases} P_{i,1} = \cos \theta_i \\ P_{i,2} = \sin \theta_i \end{cases} \quad (4.9)$$

在神经网络预测中，通常需要明确误差计算方式和误差种类。FoV 预测较为特殊的一点在于，我们可以忽略小误差，更注重于控制大误差。原因在于当发生小误差时，基于 tile 传输中多余的高质量 tile 部分（如图 4.58 等）大概率可以覆盖小误差对应的 FoV 范围，不会丢失任何高质量像素。因此，99% 和 99.9% 正确时的误差是此预测中是更为有用的指标。已有研究证明这种指标可以进一步提高 5% 的预测性能。具体而言，该方法首先根据初始数据训练神经网络。其次，收集训练错误，并通过对具有大训练误差的数据进行过采样来构建更新的数据集。最后，基于更新的数据训练增强的神经网络。

在训练或预测完成后，坐标数据仍要化为具有实际意义的欧拉角作为后续处理的输入。

Matlab 等软件中存在 atan2 函数，通过两个坐标值共同决定反正切角度，输出范围为 -180° 至 180° ，具有完备性，具体转化关系如下所示：

$$\theta_i = \text{atan}2(P_{i,2}, P_{i,1}) \quad (4.10)$$

除了利用头部数据的神经网络预测外，也有研究对图像/视频显著性检测结果与头部运动的关联性进行了验证。结果显示，即使在无时延、小窗口的情况下，传统显著性检测算法（如 GBVS）与视点移动的相关程度依旧很低，或者说这种关联性并不稳定。这一现象主要是由于传统算法的低鲁棒性会导致检测结果的巨大变化，这与相对平缓的头部运动是不匹配的。因而基于显著性的预测还需契合沉浸式媒体特点，更具鲁棒性的算法也有待被提出。

目前，已经有通过深度 CNN 网络来生成图像显著图的方法，并且采用预先训练的 CNN 学习图像特征，这些特征最初用于对象检测和图像分类。

而图 4.57 给出的两个 LSTM 预测网络（长短期记忆网络，适用于从时间序列的视频帧中学习有用信息和长期依赖关系）则再将显著性、视频帧特征和已观看 tile 等数据同时作为输入，最终以预测窗口中未来 n 个视频帧的 tile 观看概率作为输出。其中， F_f 为帧 f 的各

类特征， $p_f^t \in [0,1]$ 为帧 f 中 tile t 的预测观看概率，而帧 f 的所有 tile 概率为 P_f 。

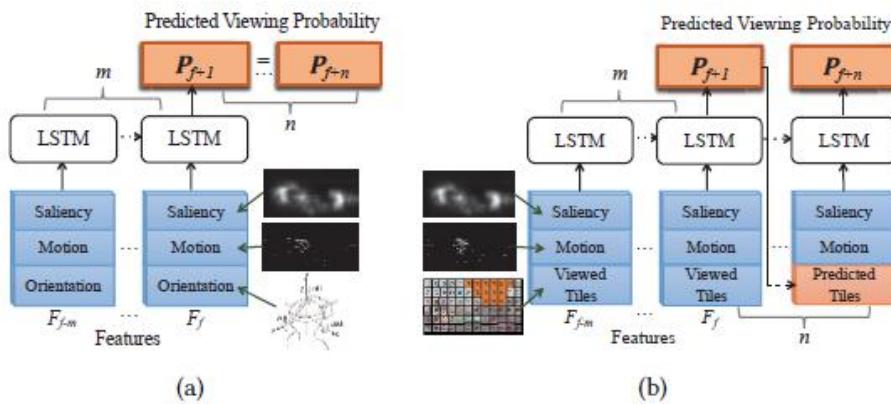


图 4.57 基于 LSTM 的 FoV 预测

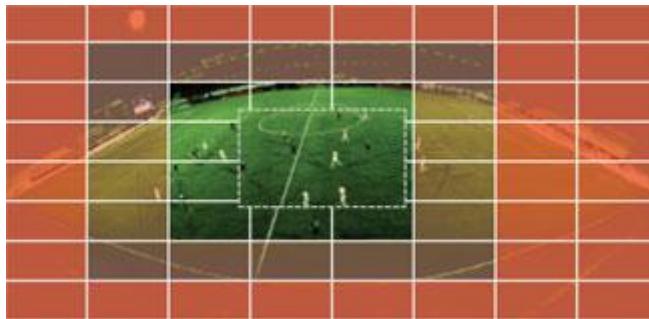


图 4.58 “金字塔”法示意图

4) “金字塔”法

“金字塔”也是一种较为复杂的方案，其根据与视点的距离，以逐渐降低的质量智能地选择质量（图 4.58），一定程度上可以改善用户体验。这种方法引入了在 $[0, 1]$ 范围内变化的优先级 (p_i) 标志，其中 0 代表最重要，1 代表最不重要。根据重要性可以获取相应的质量。然而，如果仅仅由重要性决定 tile 质量，最终可能会产生大量高质量的 tile。因而在

此方法中存在一种最高质量等级 (q_{\max}) 作为限制条件。该数值取决于高优先级 tile 的数量, 如式 (4.12) 所示。这里令 q_H 为当所有 tile 均用于视野内虚拟视图时的质量级别。

$$q_{\max} = (\sum_{i \in T} b_i / N) q_H \quad (4.11)$$

$$T_i = \begin{cases} q_{\max}, & b_i = 1 \\ q_{\max} + p_i(n - q_{\max} - 1), & b_i \neq 1 \end{cases} \quad (4.12)$$

其中, n 是质量等级的数量。得到 q_{\max} 后, 我们计算式 (4.12) 中 tile i 邻域 (N_i) 的占用率, 定义为 p_i , 如式 (4.13) 所示。从式中可以看到有多个可调参数。一个是 q_H , 它决定给定缩放级别的质量。第二个是邻域本身的选择, 可以通过 α_j 的权重来确定。我们可以使权重为各向同性或各向异性。鉴于用户更多是进行平移而不是倾斜运动, 各向异性权重可以产生与各向同性权重相似的性能, 同时消耗更少的带宽。

$$p_i = 1 - \frac{\sum_{j \in N_i} \alpha_j b_j}{\sum_{j \in N_i} \alpha_j} \quad (4.13)$$

4.4.4 基于多种分辨率和多分块的 CDN 分发方案

CDN 即内容分发网络, 其基本思路是通过网络流量和各个节点的负载情况以及对用户请求的响应时间重构链接用户到最近的服务节点上。提升内容传输的速度和稳定性, 具备高吞吐量和速度的 CDN 对于提升沉浸感的全景视频体验是十分必要的。

超高分辨率全景视频比普通平面视频的码率高出 10 倍以上, 基于块划分的编码方式也会导致编码文件比普通平面视频高出数十倍。如果采用传统分发方式, 寻址块文件全部传输将对 CDN 形成很大的压力。为了解决这些问题, 可以采用基于分块的优化分发方法, 该方法的主要思路如下。

- 对超高分辨率全景视频按照 MCTS 进行编码, CDN 只发送用户请求的分块视频, 降低全景视频网络传输的浪费, 并减轻对 CDN 的冲击。
- 提供多种分辨率格式给终端自适应请求, 网络拥堵、质量下降时, 终端请求低分辨率视频减少卡顿现象, 避免视频流畅度受到影响。
- 采用多块预拼接技术, 通过将用户索引的分块进行预拼接, 减少查询和读取 CDN 服务的次数, 因为这种频繁小文件查询和读取对 CDN 性能影响非常大。
- 引入大数据处理和深度学习等技术, 获取用户观看的兴趣区域, 并预估用户的观看内容, 提前准备需要发送的内容, 降低发送与请求之间的时延。

利用合适的预处理技术以及针对性的优化 CDN 的分发方法, 现有 CDN 具备处理 8K 分辨率全景视频的能力。通过结合新兴存储和处理技术, 处理更高分辨率的视频分发在技术上也是完全可能的。

超高清全景视频的混合分辨率显示

部分解码超高分辨率全景视频主要受限于解码器的解码能力与网络传输性能, 为了获取最佳的体验效果, 可以建立如下的目标函数:

$$\max f(x) \quad s.t. x \in \Omega \quad (4.13)$$

其中, x 表示解码的分块, Ω 表示超高频视频, 比如 8K 视频划分的全部分块。 $f(x)$ 表示基于 FoV 的呈现方式, 该函数值越大, 表示通过解码这些分块获取的用户体验越好。为了获取目标函数的最优值, 可以从如下 4 个方面考虑:

- 在解码条件允许的情况下, 获取更多 FoV 内高清晰分块;
- 在网络允许的条件下, 解码高清分块用于 VR 显示;
- 在传输能力和解码能力相同的条件下, 采用合适的补偿帧技术, 降低头部运动到显示的延迟;
- 在传输能力和解码能力相同的条件下, 通过合适的 Unwrap 方法及几何失真校正方法, 呈现失真度更小的画面。

在使用 8K 和 4K 两种分辨率视频的条件下, 图 4.59 展示了利用 3 种解码策略得到的结果。图 4.59 (a) 表示全部采用高分辨率解码得到的 FoV 显示内容, 图 4.59 (b) 为采用了混合分辨率解码得到的 FoV 显示内容, 图 4.59 (c) 为全部采用低分辨率分块解码得到的 FoV 显示内容。在观看过程中, 图 4.59 (a) 的用户体验明显高于图 4.59 (b), 但是图 4.59 (b) 的体验又优于图 4.59 (c)。这说明在网络环境和解码能力都允许的情况下, x 采用全部高分辨率分块, $f(x)$ 尽可能全部解码高分辨率分块可以获取最佳的用户体验。如果二者不能同时满足, 比如网络条件不满足时, x 可以选择部分高分辨率分块, 部分低分辨率分块传输, $f(x)$ 混合解码多种分辨率得到的用户体验也将明显高于全部传输和解码低分辨率 tile 方案的用户体验。



(a) 全部采用高分辨率分块
解码得到的 FoV 结果
(b) 左上方采用低分辨率分块
解码, 其余部分采用高分辨率
分块解码得到的 FoV 结果
(c) 全部采用低分辨率分块
解码得到的 FoV 结果

图 4.59 基于不同分辨率分块解码显示的结果

4.4.5 视频观看行为分析及显著性检测

新兴沉浸式系统提供的体验本质上与传统的广播, 电视或戏剧都不同, 为许多研究领域开辟了新的方向。然而目前, 用户探索沉浸式 VR 环境的视觉行为并未得到很好的理解, 也没有完善的统计模型来预测这种行为。实际上, 沉浸式媒体的诸多问题都会涉及到视觉行为。例如, 如何设计 3D 场景? 如何在虚拟环境中吸引用户注意力? 是否可以预测视觉探索模式? 如何有效地压缩 VR 内容?

因而, 了解用户如何探索虚拟环境至关重要。Vincent Sitzmann 等人于 2018 年公开了一项较为完整、全面的研究, 包括对用户行为的记录, 视觉行为的分析, 现有显著性检测方法的评估以及显著性检测应用场景的扩展四个方面。

1) 用户行为记录

该团队记录的数据集包含多种条件下观看全景视频时用户的头部方向和注视方向, 这些数据集形成了视觉行为统计分析的基础, 是显著性预测的实际对比数据, 也是更高级别应用

的显著性参考集。

观看条件

该研究使用了 22 个高分辨率全景视频（图 4.60 为 22 个视频的一部分），并记录了用户在三种不同条件下的观看情况：佩戴 HMD；佩戴 HMD 坐在非旋转椅上，使其难以转身；坐在桌面显示器前。在桌面条件下，场景是单视场的，用户使用鼠标进行视角切换。对于每个场景，团队测试了四个不同的起点，间隔为 90 度，总共 264 个条件。选择这些起始点是为了覆盖整个水平范围。同时还有四个固定纬度的随机起始点以限制条件数量。

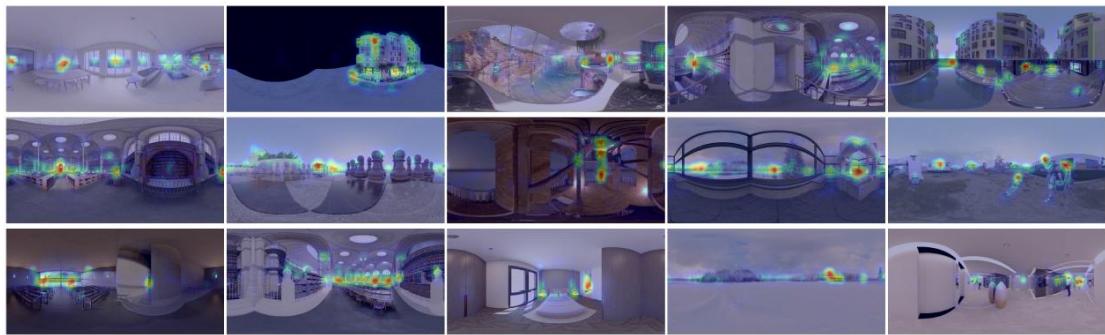


图 4.60 部分测试视频

参与者

实验总共记录了 122 名用户（92 名男性，30 名女性，年龄 17–59 岁）。HMD 坐姿状态的实验由 47 名测试者（38 名男性，9 名女性，年龄 17–39 岁）完成。测试前要求用户首先进行立体视觉测试以量化他们的立体视敏度。对于桌面实验，团队招募了 44 名额外的参与者（27 名男性，17 名女性，年龄 18–33 岁）。所有参与者的（矫正）视力均正常。

测试程序

测试使用 Oculus DK2 显示所有 VR 场景，配备有以 120 Hz 记录的 pupil-labs 立体眼动仪。DK2 提供 $95^\circ \times 106^\circ$ 的视野。Unity 引擎用于制造所有场景和记录头部方向，而眼动仪在单独的计算机上收集视角数据。用户在测试时还需戴上耳罩以避免听觉干扰。场景和起点是随机的，同时确保单次测试中每个用户只能从一个随机起点观看相同的场景。每个用户显示 8 个场景，并在 30 秒内向用户显示特定条件下的各个场景。

对于桌面条件，用户距离 17.3 英寸显示器 0.45 米，分辨率为 1920x1080，覆盖 $23^\circ \times 13^\circ$ 的视野。对应的图像查看器显示了一个 $97^\circ \times 65^\circ$ 直线投影的全景窗口。这种情况只收集视角数据，因为用户很少会有头部运动。取而代之的是使用虚拟相机在全景中的放置等效为头部位置。

2) 视觉行为分析

用户间观看行为的相似性

首先评估用户之间的观看行为是否相似。该团队通过接收器工作特性曲线（ROC）计算用户之间的一致性指标。图 4.61（左）展示了所有 22 个场景的平均 ROC，并与每个场景的 ROC（浅灰色）相比较。这些曲线快速收敛到 1 表示用户间的一致性很强，因此行为相似。约 70% 的曲线在最显著的前 20% 区域内就已接近 1。

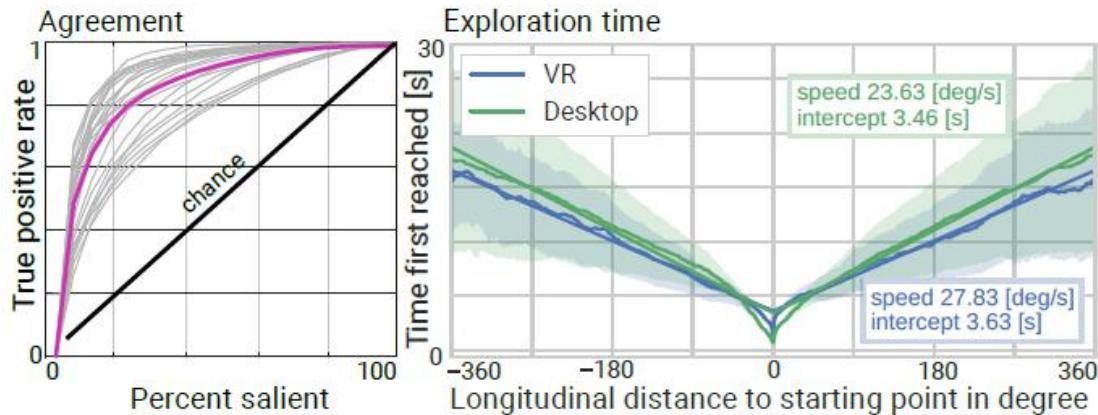


图 4.61 左：平均（红色）和各场景（浅灰色）的人类行为 ROC 曲线。快速收敛到最大值表明用户行为之间的强烈一致性。右：Exploration Time 表示与起始点达到特定经度距离时的平均时间。

不同情况下的观看行为

该团队进而分析用户在不同观看条件下是否会改变行为。为定量评估显著图的相似性，其使用了 Pearson 相关 (CC) 分数，是显著图预测中广泛使用的指标。当比较 VR 和 VR 坐姿条件时，中位 CC 得分为 0:80；而比较 VR 和桌面条件，得到 0:76 的分数，确实存在高相似性。后者是一个重要结果：由于桌面实验更容易控制，因此可以使用此类实验来收集行为数据集。

视点偏向性

有报告指出，当观看传统图像时，人类视点位置会偏向中心。相同的问题在 VR 中也需要被解答。对此，团队计算了 22 个显着图的平均值，得到的数据表明用户倾向于注视全景图赤道附近的内容。图 4.62 显示了 VR 和桌面条件下的平均显着图，以及纬度分布及其参数的拟合，可以看到两种条件的平均值几乎相同。

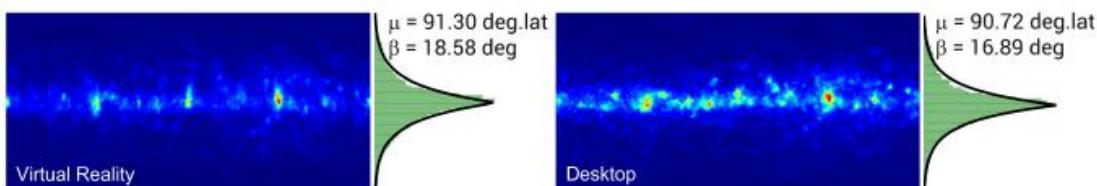


图 4.62 VR（左）和桌面（右）条件下所有场景的平均显著图



图 4.63 具有最低（左）和最高（右）熵的显著图

场景内容对于观看的影响

分析观看行为时的一个基本问题是场景内容的潜在影响，此分析一定程度上可以帮助解

决用户视点预测的难题。

团队根据场景中显著区域的分布，以显著图的熵方式表征场景内容。高熵是由分布在整个场景中的大量显著物体产生的，导致用户的注视点分散在整个场景中；低熵是由一些集中显著物体产生的。图 4.63 显示了数据集中具有最低和最高熵场景的显著图。



图 4.64 显著区域计算。左：完整显著图。右：通过对场景的前 5% 显著像素进行阈值处理得到的显著区域（黄色）。

A. 用户行为指标

以客观指标衡量观众行为并不是一项简单的任务。首先，团队将显著区域定义为场景中前 5% 显著像素。图 4.64 显示了显著图和由此标准计算得到的显著区域。然后结合 Serrano 等人最近提出的三个指标（到达显著区域的时间（timeToSR），显著区域的注视百分比（percFixInside）和注视数量（nFix）），提出了第四个契合 VR360 视频的指标：

收敛时间（convergTime）：对于每个场景，团队在不同的时间步长获得每个用户的显著图，并使用完全收敛的显著图来计算相似性（CC 得分），并绘制 CC 得分的变化趋势，进而计算该曲线下的面积。该指标表示显著图的时间收敛性；它与观看轨迹图收敛到实际显著图所需的时间成反比。

B. 分析

经测试，团队发现场景熵对 nFix, timeToSR, percFixInside 和 convergTime 均有显著影响。具体而言，对于具有低熵的场景，timeToSR 较低。这可能是违反主观直觉的，因为高熵场景包含更多的显著区域，更容易快速达到；有趣的是，结果表明用户在低熵情况下能更快地探索场景，快速丢弃非显著区域，并且注意力会更快地指向少数的显著区域。convergTime 指标结果进一步支持了这一结论，该指标表明低熵场景确实收敛得更快。nFix 和 percFixInside 的测试结果同样可得到相似的结论。

3) 显著图预测

本小节中，团队将现有模型用于沉浸式场景。这是合理的，因为已经存在许多用于桌面条件的显著性预测方法，并且该领域的进展可以直接转移到 VR 条件。但在 VR 条件下，主要出现了以下两个问题：(i) 映射 360 全景至 2D 图像时，球体到平面的投影扭曲了内容；(ii) 头眼交互可能需要在 VR 显著性预测中被特别注意。以下便解决了这两个问题。

哪种投影最好？

在对球形全景应用传统显著性预测方法之前，必须将图像投影到平面上。不同的投影会导致不同类型的失真，这些失真会影响显著预测因子。例如，对于经纬图投影，极点附近会出现大的扭曲。立方体投影会导致立方体面之间出现不连续的现象。又或者，可以从全景图中提取较小的图像块，将预测应用于每个小图像块，最终将结果拼接在一起并混合到全景图中。这种基于小块的方法将引入最少量的几何失真，但它也是计算成本最高的方法，并且它

放弃了显著性预测的全局背景。

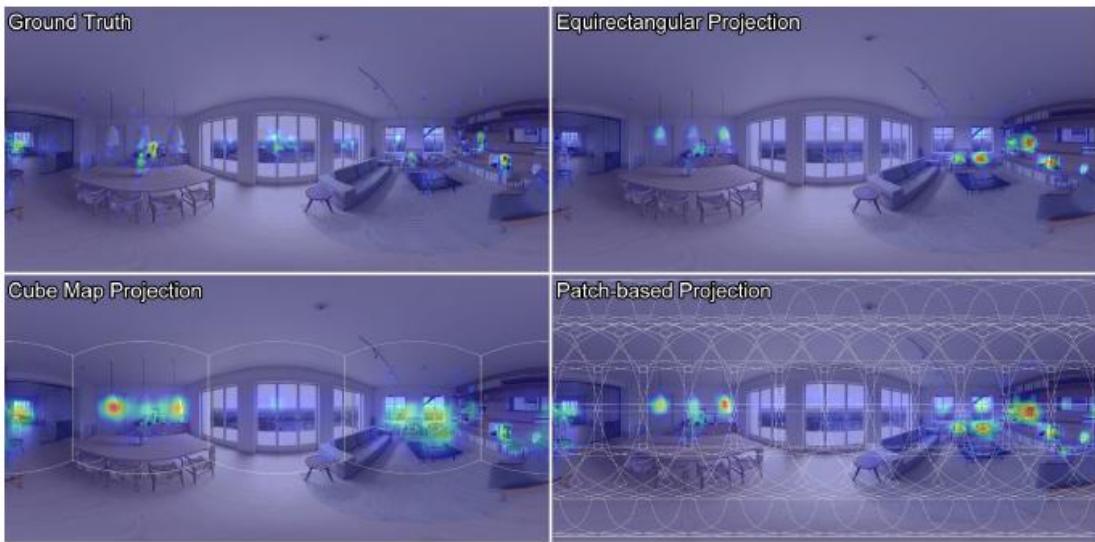


图 4.65 使用不同投影方法的显著性预测效果比较。在加入赤道偏差后，所有三种投影方法都产生了与示例相当的显著图。

表 4.4 有/没有赤道偏差时，三种投影方法的定量评估

	经纬图投影	立方体投影	小块法
无赤道偏差	$u = 0.48$	$u = 0.37$	$u = 0.43$
有赤道偏差	$u = 0.50$	$u = 0.44$	$u = 0.49$

图 4.65 和表 4.4 使用了上述三种投影方法定性和定量地比较显著性预测效果。对于每种投影，团队使用最先进的 ML-Net 显著性预测法计算显著图，然后将其分别乘以之前小节中得出的纬度赤道偏差。图 4.65 显示了在应用赤道偏差之后，三种不同球面投影预测的显著图。此外，团队还于表 4.4 中比较了三种投影方法所有 22 个场景的平均 CC 得分。定量而言，带有赤道偏差的经纬图投影图计算的显著性不仅表现最佳，而且也是三种方法中最快的。对于经纬图投影而言，应用赤道偏差的收益可能小于其他两种投影，因为极点处的失真会导致在极点处预测的显著性低于立方体图和基于小块的方法。

表 4.5 使用简单赤道偏差 (EB) 和两种最先进模型预测显著性的定量比较

	EB	ML-Net+EB	Sa1Net+EB
VR 条件	$u = 0.34 \pm 0.13$	$u = 0.49 \pm 0.11$	$u = 0.47 \pm 0.13$
桌面条件	$u = 0.37 \pm 0.11$	$u = 0.57 \pm 0.11$	$u = 0.52 \pm 0.12$

哪种预测方法最好？

这里主要从数量和质量上评估现有的几种预测因子。表 4.5 列出了 VR 和桌面条件下所有 22 个场景的 CC 分数的平均值和标准偏差。从这些数据中能够分析预测方法的优异程度和一致性。这里以赤道偏差本身作为基准，同时测试 MIT benchmark 中排名最高的两个模型：MLNet 和 Sa1Net。从表中可以看到两个高级模型的表现非常相似，均比赤道偏差更好。同时还可以看到，这两种模型在桌面条件下的预测效果比 VR 条件更好。因为是桌面条件原本就是这些模型训练的条件。图 4.66 中定性比较了 VR 条件下记录的三种场景的显著图。

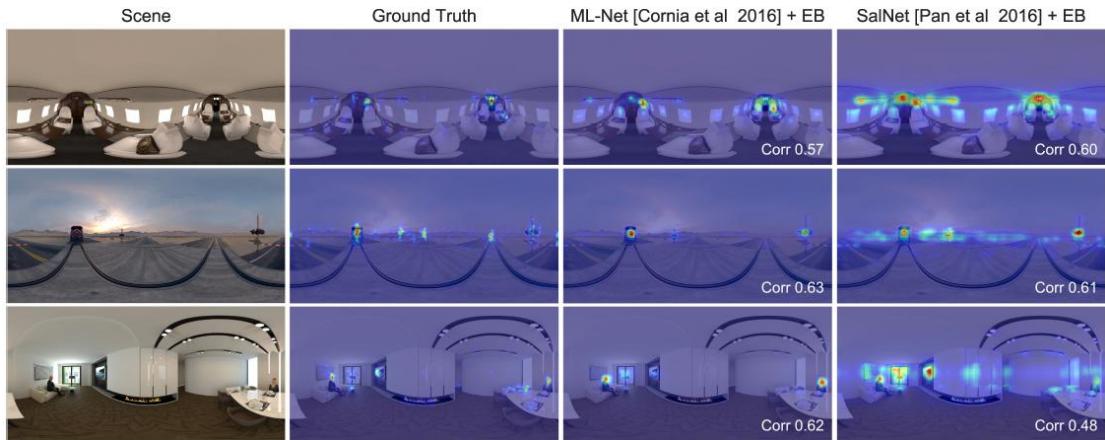


图 4.66 全景图显著性预测，这里使用基于小块的方法完成投影。通过将预测显著图与前一部分中导出的纵向赤道偏差 (EB) 相乘，团队实现了实际（中间左侧）和预测显著性（右侧）之间的良好匹配。需要注意的是，此过程可应用于任何预测方法。

4) 显著性预测的应用场景

主要有以下四种：

- VR 视频片段的自动对齐；
- 全景缩略图表示；
- 全景视频简介的自动生成；
- 基于显著性的 VR 视频压缩。

4.4.6 “全景声巨幕影院”的技术创新和变革

2018 年，大朋 VR 在自主研发的 VR 技术上再次进行了创新和优化，在国产芯片基础上，极大地提升了 VR 体验质量。

中国“芯”，新起点

现有市场上 VR 一体机多基于美国高通或韩国三星的芯片，大朋最早的一体机 M2 也不例外，使用的也是三星芯片。

而在如今“声援”国产芯片的大背景下，大朋提前选定了国内全志的 VR9 芯片。大朋曾向全志建议把 ATW 等 VR 算法操作从 GPU 中释放出来，打造专有硬件模块提高效率。目前，全志已经为全球 VR 市场发布了第一款专用芯片 VR9，其定位正是给用户提供极致的 VR 影音体验。尺有所短寸有所长，VR9 的影音解码是强项，但以此就牺牲了游戏 GPU 能力。

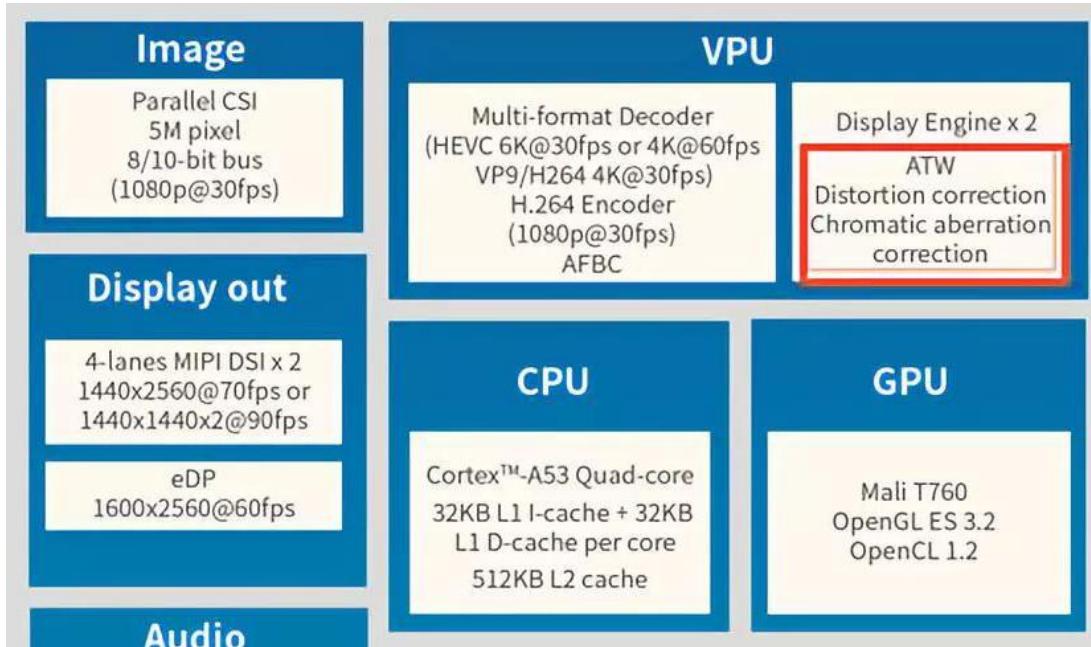


图 4.67 全志 VR9 框架图

在 VR9 国产芯片设计的基础上，大朋对底层至上层均进行了核心优化，接下来就将对 VR 巨幕影院的技术创新进行介绍。

VR 流水线：从渲染到人眼

要想真正了解 VR 技术的本质，首先应知道 VR 世界中一个物体是如何被渲染并最终进入人眼的。

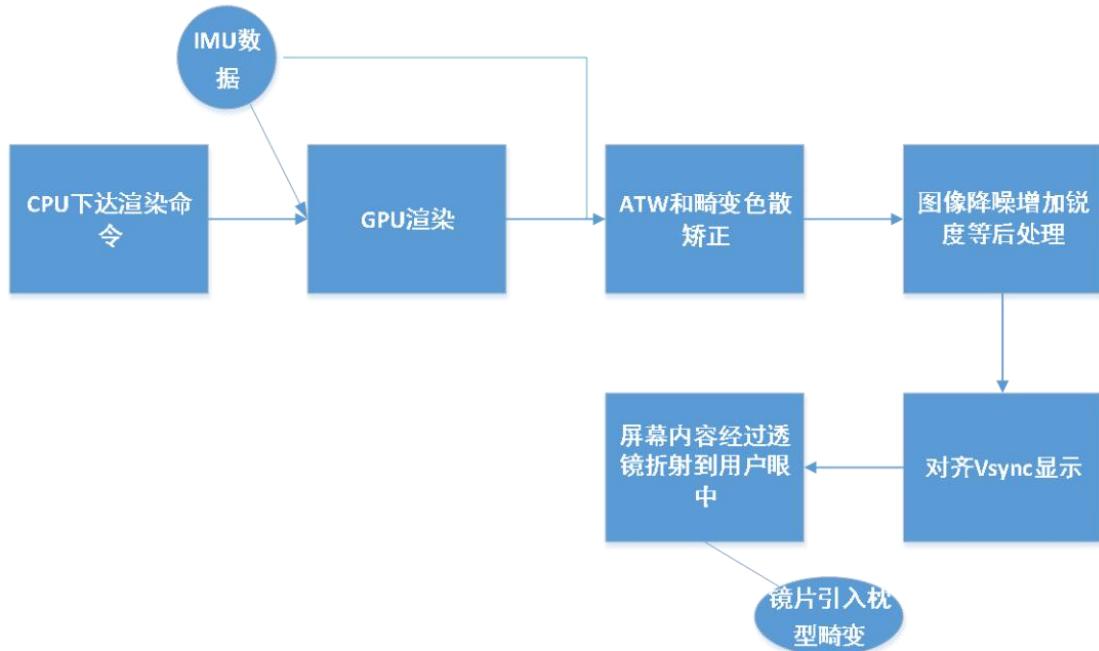


图 4.68 VR 物体进入用户眼中的历程（流水线）

VR 系统本质上是一个异构计算系统，内部的 CPU, GPU, Display 等硬件模块始终协同并行工作。VR 世界中的每一个物体从第一个模块开始，在整个流水线上一步步推进，最终

进入用户眼中，如何提高流水线中每一步的效率和并行度是 VR 系统高效运转的关键。

1) 巨幕影院渲染算法优化

由于设计时受到功耗和芯片面积的限制，移动端 GPU 性能参数，不管是 FLOP 还是内存带宽都大大低于同级别的 PC GPU，比如 Nvidia 的 PC 端 GPU GTX 650 和移动端 GPU Tegra K1，虽然都来自于 Kepler 架构，出现的时间几乎相同，但前者的内存带宽是 80G/s，后者的只有 18G/s。对于用户来说，这个差别意味着移动端的 VR 应用和实现不可能采用和 PC 系统一样的方法。而对于 VR SDK 的提供商来说，只能想办法提升移动平台上 GPU 的利用率。在这个背景下，能够挤掉 CPU 和 GPU 之间泡沫，提高两者运行并行度的 Adaptive Queue Ahead 技术应运而生。

以前的 VR 世界中，CPU 总是在 VSync（垂直同步）到来才开始下达渲染命令给 GPU（如下图），对于较重的 GPU 任务，很可能无法在当前 VSync 剩余时间中完成，后果就是应用的 FPS 下降，最终用户体验到应用或者游戏卡顿，显示“鬼影”以及眩晕。

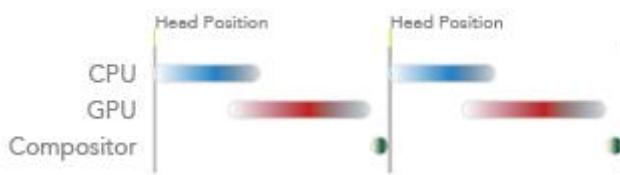


图 4.69 不带 Queue Ahead 的渲染

Oculus 最早在 PC 端的 Rift 上提出所谓的 Adaptive Queue Ahead 技术，使得 CPU 不用等待 Vsync 的到来，而是通过预测，在 VSync 到来之前几毫秒内开始下达渲染指令给 GPU，让 GPU 有更多的时间执行任务，有效提高 VR 应用的 FPS，产生更好的用户体验。

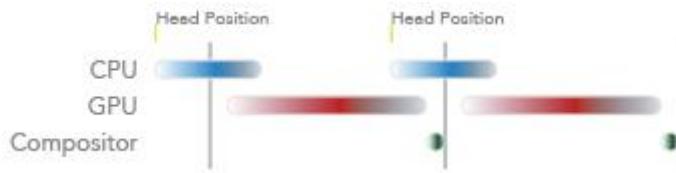


图 4.70 带 Queue Ahead 之后的渲染

大朋则首次把来自 PC VR 端的技术引入到 VR 一体机的世界，让以前运行卡顿的应用流畅起来，还给用户一个平滑，沉浸和画面精致的 VR 世界。不过，考虑到 PC 平台和一体机平台之间的计算能力差异，仅这一个优化还远远不够，于是大朋又通过名为“Hidden Mesh”的技术进一步提高 GPU 的渲染效率。

在 VR 头盔的光学视场中，由于镜片结构和人眼特点，图像中某些区域人眼是无法看到的，在 VR 图像渲染中被称为 Hidden Area（如下图中红色三角覆盖的地方，人眼其实无法看到）。

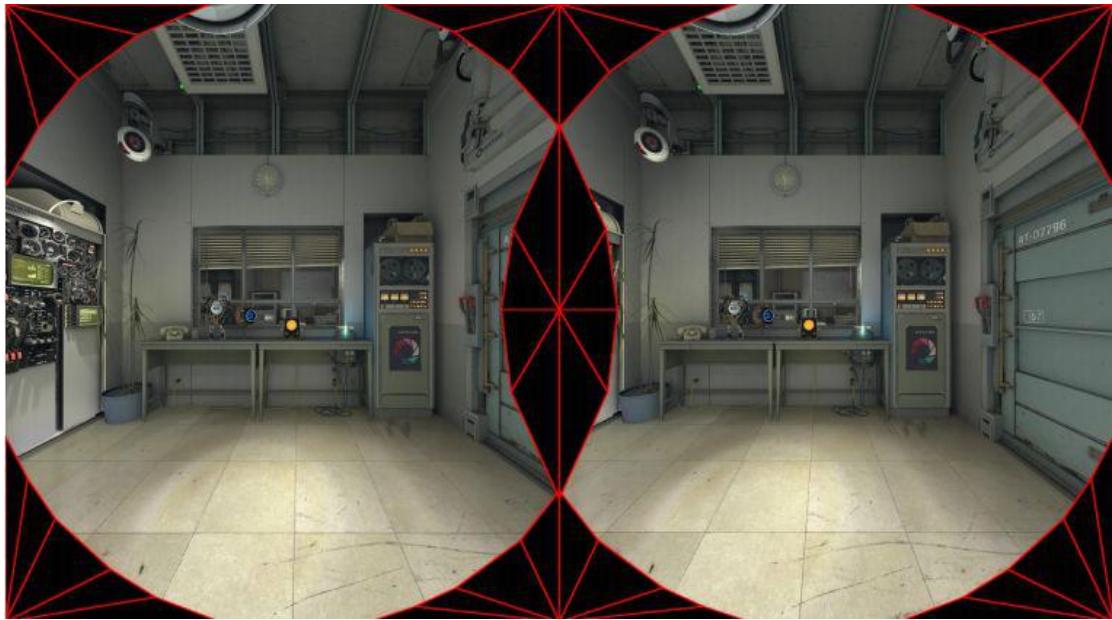


图 4.71 Hidden Mesh 技术

大朋巨幕影院的图形渲染则巧妙地利用到了这点，通过利用特殊绘制的 Hidden Mesh（隐藏网格），有效地降低了 GPU 的渲染工作量，提高了 CPU, GPU 并行度和 GPU 渲染效率。而以其他系统作为对比，如 Oculus Go，也许出于其他的考虑，它并没有采用 Hidden Mesh。下图中红框是 Oculus Go Home 中用户能够看到的部分，红框之外圆圈之内的内容用户通过透镜和镜片并不能看到。

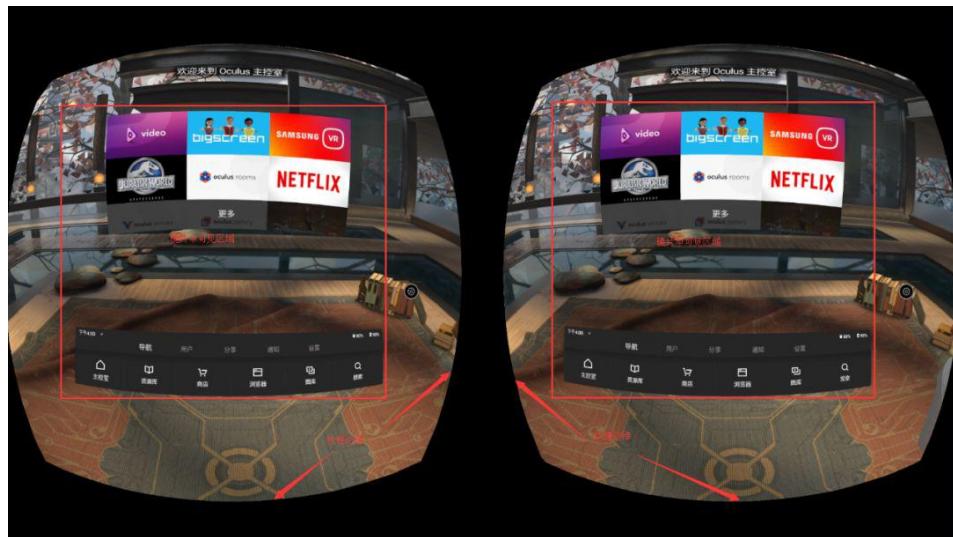


图 4.72 Oculus Go Home

除此之外，还需要有效减少用户佩戴时的眩晕感。人类的身体并非是天生为 VR 而设计的。通过 VR 设备对感官进行的人工刺激会破坏生物机制的运作，这些机制经历了数亿年时间在自然环境中演变而来。同时这也向大脑提供与现实体验不完全一致的信息。在部分情况下，我们的身体可能会适应新的刺激。但在其他情况下，我们的身体会产生眩晕和恶心等症状，一部分原因是大脑比平常更高速地运转，以理解这类刺激。已知的产生眩晕的原因除了显示分辨率/刷新率不足，前庭和视觉系统冲突，虚拟世界中比例失真外，MTP 延时过大也是其中的一个因素。

在之前也提到过，一般情况下 MTP 延时大于 20 毫秒便会导致用户体验到较为明显的眩晕。

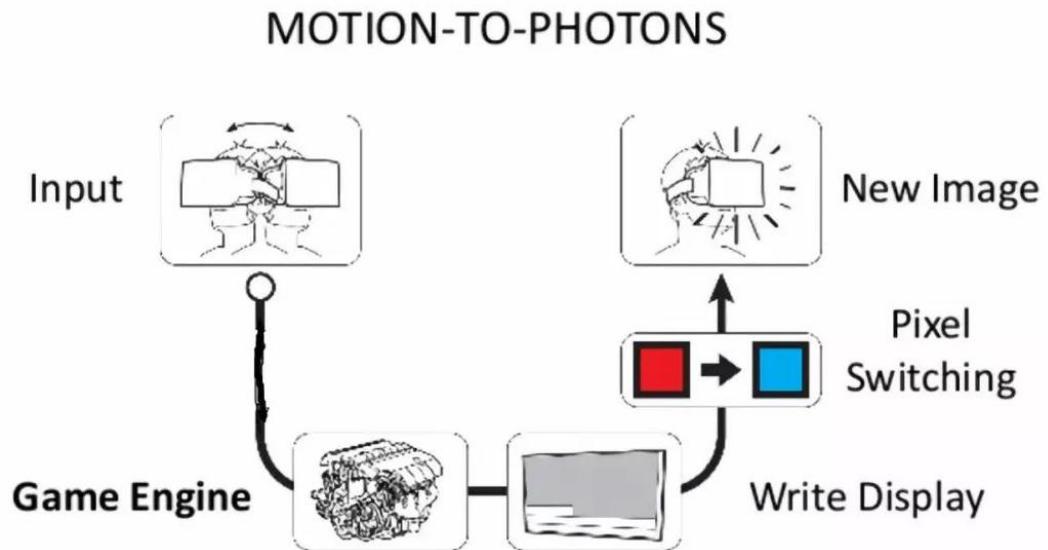
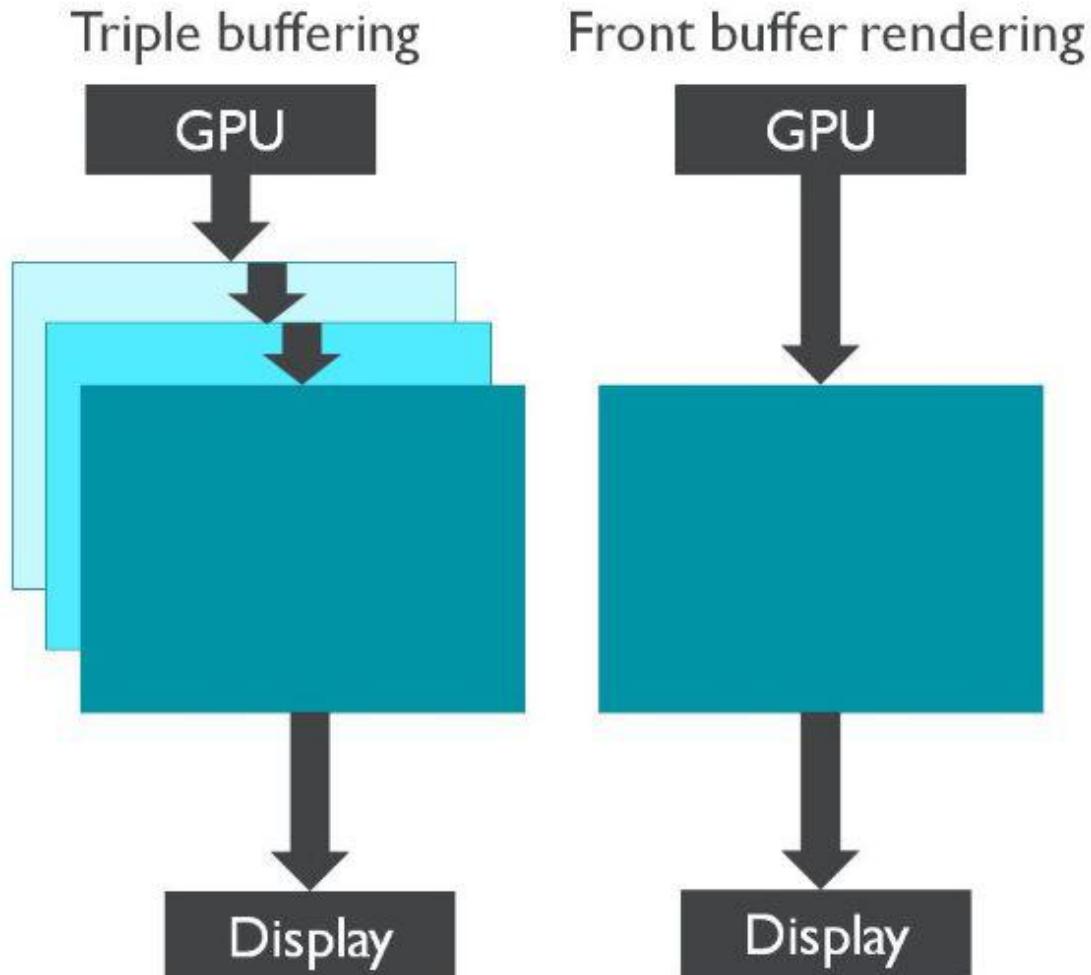


图 4.73 MTP 延迟

而现有市面上的 VR 一体机无一例外都是基于 Android 系统。为了提高手机和平板电脑上显示的平滑性，传统的 Android 系统均采用双显示缓冲或者三显示缓冲。但是，这个机制让 VR 应用无法知道指定的图像什么时候能够显示在头盔屏幕上，反而加大了 VR 一体机的 MTP 延时，让用户体验到更多的眩晕。大朋则对此进行了硬件结构和算法优化，使得 Front Buffer Rendering（前屏渲染）成为可能：流水线中只采用了一个显示缓冲，最大程度上减少了 MTP 延时，提供给用户更好的视觉体验。



Reduces latency by omitting additional buffer passes

图 4.74 Front Buffer Rendering

2) 显示优化

除了渲染性能，显示清晰度一直是判断 VR 头盔优劣的另外一个重要指标，不过，没有所谓显示优化的“银弹”能一招制敌，清晰度的提升来自于各个模块的综合效果。大朋 VR 则对其中的各部分均做了不断的迭代和改进，从而产生了良好的效果。

首先，GPU 渲染出来的画面应是清晰的。但是，计算机渲染的场景从三维空间的角度看是连续的，经过光栅化之后最终显示在屏幕上的二维的图像本身却是离散的，这导致非完全垂直或者非完全水平的边上出现锯齿。

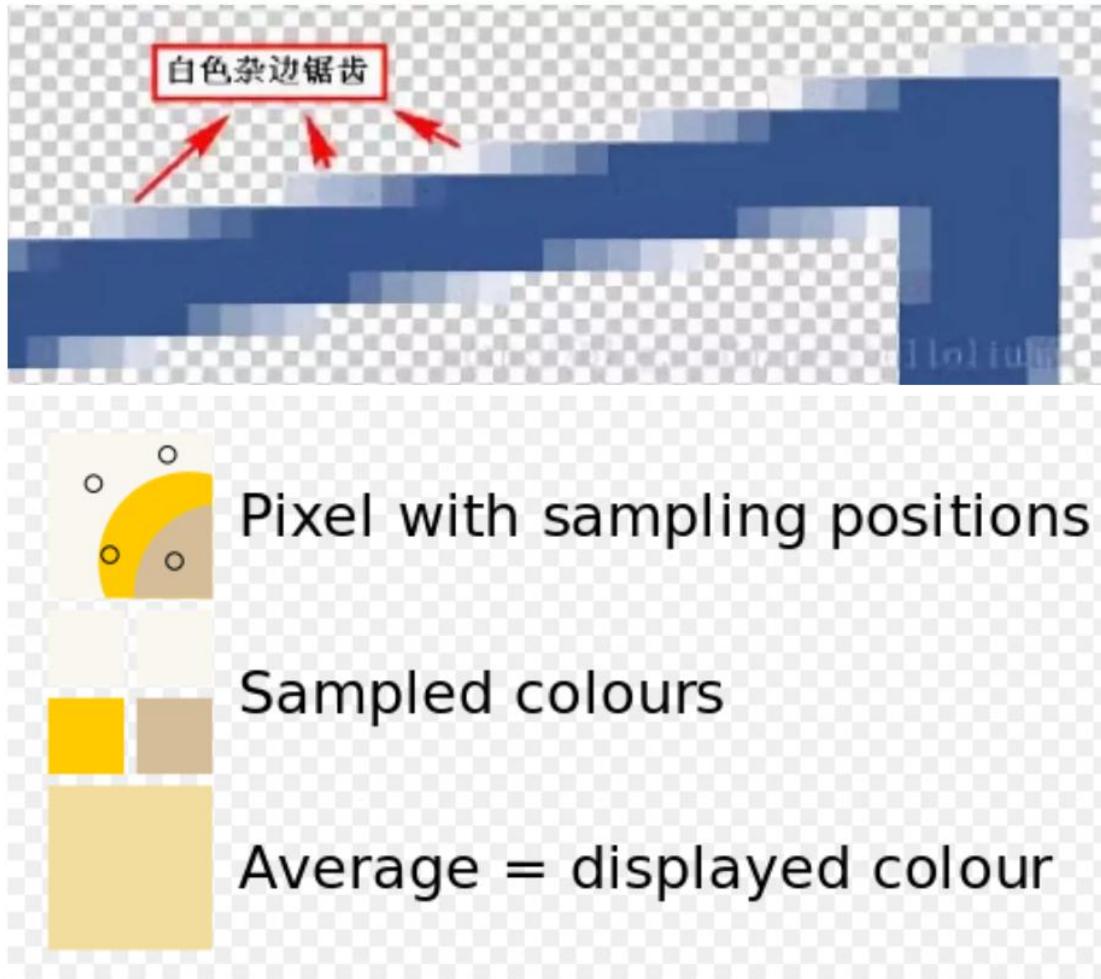


图 4.75 SSAA/MSAA 减轻锯齿

抗锯齿最直接的方法是 SSAA (Super Sampling Anti Alias) 和 MSAA。具体的思想都是先把物体渲染到比屏幕分辨率大 (比如 4 倍) 的缓冲区中, 然后再降采样到和屏幕分辨率一样的显示缓冲区中, 最后输出显示, 这样更多的信息被保留, 而图像物体边缘的颜色也因为混合了不同颜色采样点而消除或者减轻了锯齿。大朋巨幕影院的图形渲染实现也采用了 SSAA 和 MSAA 来抗锯齿。

然而, 优化并不会就此为止, VR 用户常常会抱怨图片或者文字闪烁。为什么我们在 PC 或者手机上看不到闪烁而在 VR 头盔中很容易看到呢? 这主要是因为用户调整了在 VR 世界中与物体的距离, 或者图像、文字本身存在缩放, 再加上透镜本身的放大作用, 用户就会观察到闪烁。

大朋采用了 MipMap 技术来防止文字和图片的闪烁。MipMap 是指根据距观看者远近距离的不同, 以不同的分辨率将单一的材质贴图以多重图像的形式表现出来: 尺寸最大的图像放在前面显著的位置, 而相对较小的图像则后退到背景区域。每一个不同的尺寸等级定义成一个 Mipmap 水平。



图 4.76 Mipmap 防止闪烁

这样，每次渲染的时候系统会找出相对当前场景最适合的图像，做最小的缩放操作或者根本无需缩放，让图像信息最大程度的保真。

3) 70HZ 显示刷新率

和 Oculus Go 一样，大朋采用了快速响应 fast-LCD 屏幕，区别在于，Oculus Go 缺省的刷新率是 60HZ（某些特殊情况可以到 72HZ），而大朋的刷新率则一直为 70HZ。

Fast-LCD 屏幕上的像素点在每个 Vsync 过程中并不是完全点亮，屏幕的余辉（Persistence）大概在 1-2ms。假设屏幕的余晖是 1ms，对于 60HZ 而言，有 6.25% 的时间屏幕上像素点是亮的，而对于 70hz 刷新率来说，就有 7% 的时间是亮的，因而大朋巨幕影院用户会感觉 VR 世界更加明亮。同时，人眼工作是在一个更高刷新率的模式，而较低刷新率的 VR 头盔也会让用户感到闪烁。

4) 显示芯片中的异步时间扭曲

在一个清晰，高刷新率的平稳世界中，常见的 VR 眩晕还会有吗？仍然有可能。带上头盔的用户会在使用过程中不停的转动，图像渲染时采用的姿态信息和图像在屏幕显示时的姿态可能完全不一样，用户也同样会有晕眩的症状。

对此，解决方法是在图像帧扫描到显示器之前进行再一次的调整：根据最新的预测姿态更新图像，这被称为 Time Warping（时间扭曲）或者 Reprojection（再投影）。如果在实现中渲染的线程和做扭曲的线程是不同线程的话，又被称为 Asynchronous Time Warping（异步时间扭曲）。

一般而言，异步时间扭曲（包括畸变矫正和色散矫正）在 GPU 中完成。

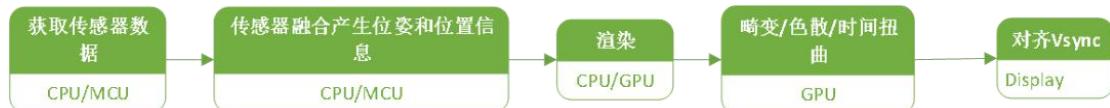


图 4.77 传统的 ATW

但是，由于 VR 游戏或者应用会在渲染环节占用大量的 GPU 资源和计算能力，会造成 GPU 不能及时完成以上任务，带来较差的用户体验，这在移动端尤为明显。大朋巨幕影院中则首次将时间扭曲/畸变矫正/色散等处理放在了独立的显示芯片中完成，减少了 GPU 负载，释放了 GPU 资源，有效提高了系统性能，也降低了系统功耗。



图 4.78 显示芯片中的 ATW

5) 图像后处理机制

目前，移动端上的大部分摄影 APP 都带有滤镜功能，而在 VR 世界中也可以产生同样的效果。这里所谓的滤镜，其实就是图像后处理。大朋巨幕影院系统中的图像后处理系统被称为 SmartColor，能够带来更鲜艳的色彩和更好的色温控制，包括如下的功能：

- (1) 自适应的细节和边缘增强；
- (2) 自适应的颜色增强；
- (3) 自适应的对比度增强和色调矫正。

经图像后处理后，画面的色彩将更加自然，脸部层次会更丰富，头发等细节显示更加细腻，如下图所示。



图 4.79 图像后处理比较(右为处理后效果)

6) 透镜设计

VR HMD 内的透镜本质上就是一个放大镜，也是 VR 中很多光学缺陷比如纱窗效应，杂散光的根源。在显示屏分辨率大致相同的情况下，VR 镜片的关注点主要有两个：透镜中心到透镜边缘的清晰度下降快慢，菲涅尔杂散光和拖影。

比如，下图被美国军方用来检测镜片各区域的清晰度。如果把图放入 HMD 中，你能清晰看到图像中间水平和垂直红线的最大刻度是多少？

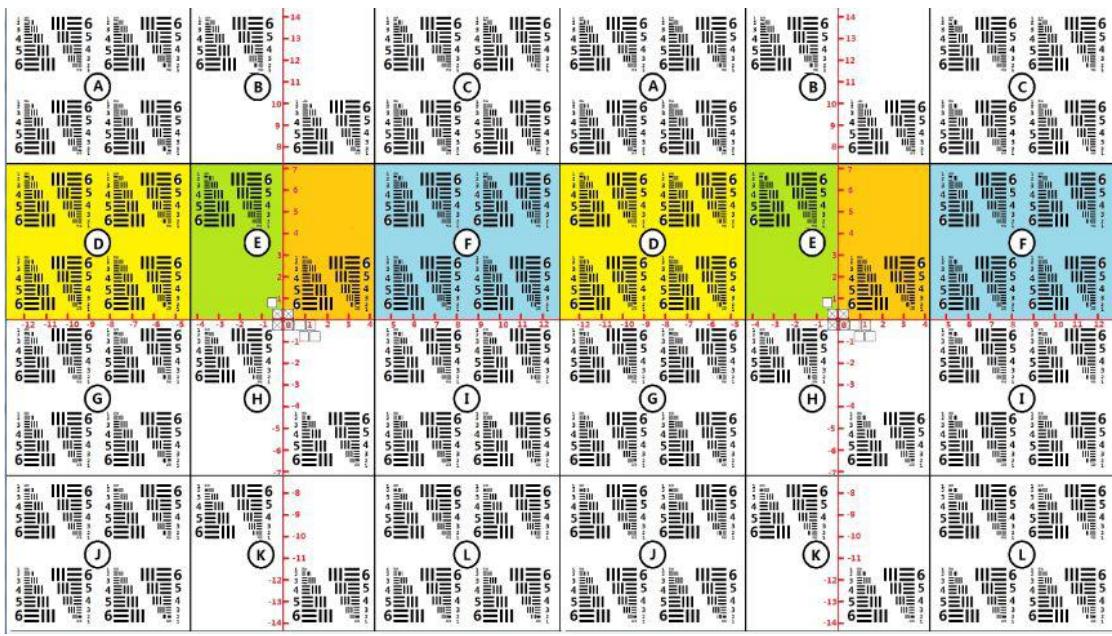


图 4.80 清晰度比较基准图

经过测试，将以上图片导入大朋工程样机和 Oculus Go 时，左右两侧能看到的最大清晰刻度分别是 11.2（大朋），11.2（Oculus Go）。从清晰度的下降程度来看，通过 HMD 看以上图片时，大朋巨幕影院和 Oculus Go 可以达到同样的边缘清晰度。

和 Oculus Go 相同，大朋巨幕影院采用了菲涅尔镜片。和非球面镜片相比，菲涅尔镜片更轻，视场角也能做的更大，长时间使用更能保护用户的眼睛，但是由于其特殊的工艺和形状，齿间的光漫反射等原因，会造成杂散光和特殊的光晕。

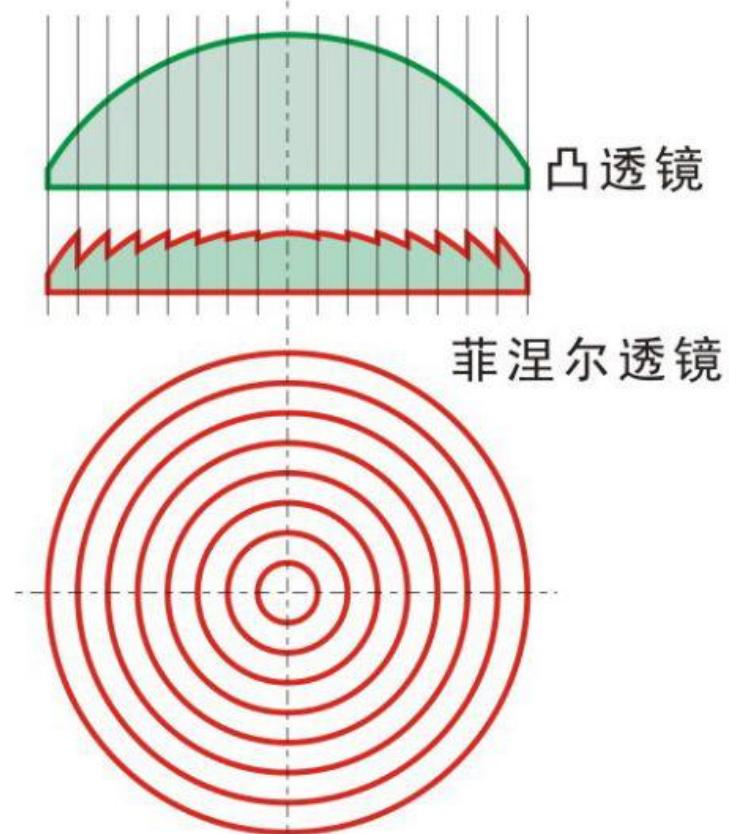


图 4.81 菲涅尔透镜外观

因而需要保留菲涅尔镜片的优点，同时补足其缺点，大朋的光学镜片进行了专门的设计优化，有效地消除了杂散光和光晕。这种优化效果在黑暗背景下由亮光形成的图案中，可以很好地被观察到。

经过测试，从 Oculus Go 中抓取到并在大朋巨幕影院 HMD 内显示的画面，视野内左下角“未安装应用”、“环境”等白色字体的拖影情况与 Oculus Go 中显示的拖影几乎相同。



图 4.82 拖影测试图

此外，在大朋巨幕影院中打开“3D 影视”→“三少爷的剑”，在影院场景中选择第 7 排，然后“关灯”，时间轴定格到 00:01:02 暂停，画面会显示下图内容，可以发现虚拟银幕以外的区域几乎没有杂光光晕，

作为对比，相同条件下 Oculus Go 中虚拟银幕以外区域的杂光光晕会略微差些。

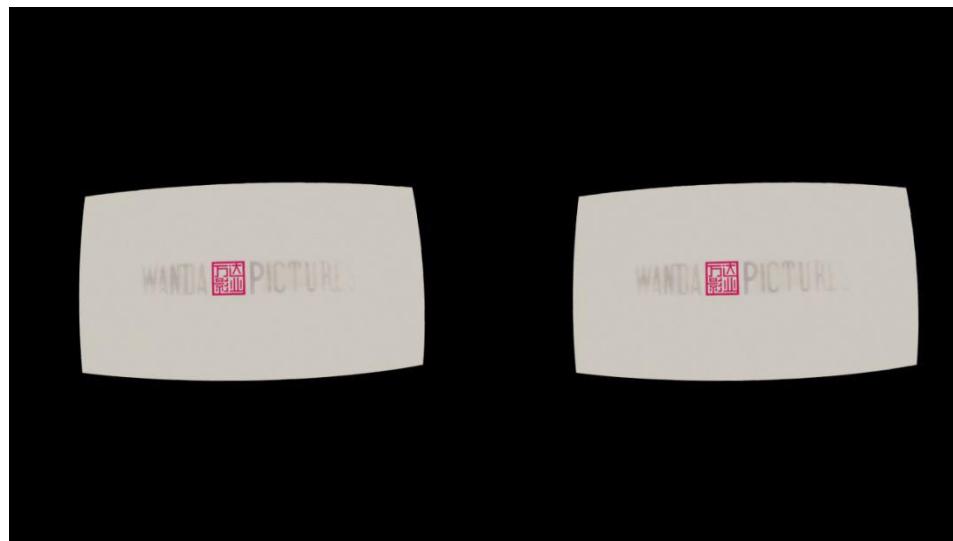


图 4.83 光晕测试

环绕立体声和定向声场

为强化沉浸感，大朋巨幕影院中加入了杜比 7.1 声道模拟算法，让用户观看视频时能体验到 360 度环绕立体声效果。

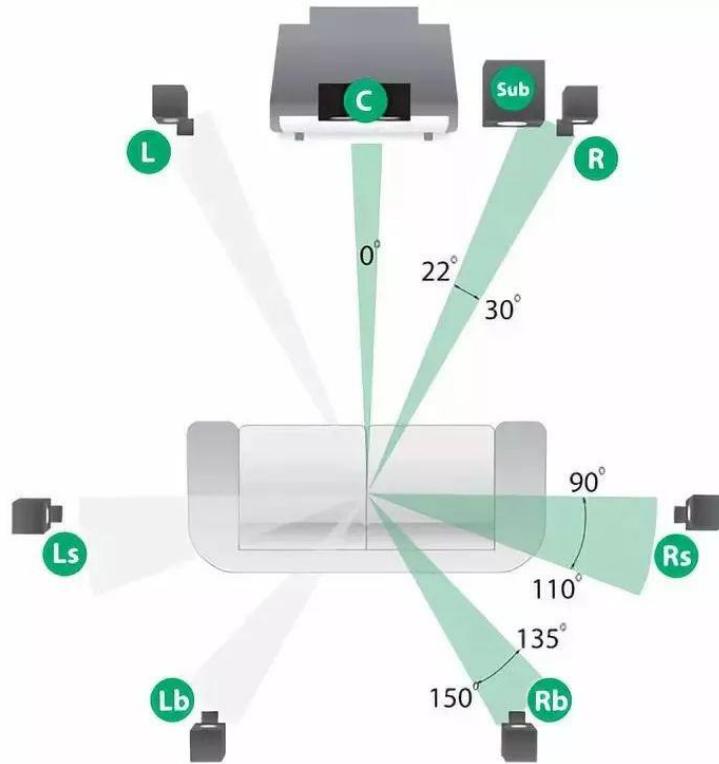


图 4.84 VR 7.1 声道

同时，为了降低周围环境对用户的影响，还实现了定向声场传播，使用者本人和周围的旁观者会听到完全不同的效果。

深度功耗优化

根据 CPU 自身的状态，大朋巨幕影院系统能够进入到 3 个不同的功耗等级：正常，待机和深度睡眠，实测观影续航能到 4 小时。

结合相应的用户操作和接近开关，大朋巨幕影院系统能够自动在不同模式之间切换，达到节电的目的。同时，根据当前 CPU、GPU 等硬件模块的负载，巨幕影院能动态调节 CPU、GPU 的频点，以满足不同使用场景的性能需求。比如当 CPU 使用率大于某一阈值时，会将 CPU 运行在更高的频点，以满足更大的性能需求；当 CPU 使用率小于某一阈值时，系统会将 CPU 运行在更低的频点，以满足更低功耗的需求。

本章参考资料：

- [1] 罗莹，宋利，解蓉，等. 全景媒体的系统架构研究综述[J]. 电信科学，2018(2).
- [2] <https://zh.wikipedia.org/wiki/MPEG>
- [3] <https://zh.wikipedia.org/wiki/MPEG-1>
- [4] <https://zh.wikipedia.org/wiki/MPEG-2>
- [5] <https://zh.wikipedia.org/wiki/MPEG-4>
- [6] <https://zh.wikipedia.org/wiki/基于HTTP的动态自适应流>
- [7] <https://blog.csdn.net/wesleyhe/article/details/6930591>
- [8] https://archive.fosdem.org/2017/schedule/event/om_gpac/attachments/slides/1886/export/events/attachments/om_gpac/slides/1886/FOSDEM17_GPAC.pdf
- [9] <https://mp.weixin.qq.com/s/H2BHUKH17ZsJEZS30ubKg>

- [10]https://mp.weixin.qq.com/s/RCb_-DhcN-6Edit5Rn37sA
- [11]<http://tech.tom.com/201807/1060863396.html>
- [12]<https://ieeexplore.ieee.org/document/8281390/>
- [13]<https://ieeexplore.ieee.org/document/8424251/>
- [14] Feuvre J L, Concolato C, Moissinac J C. GPAC: open source multimedia framework[C]// ACM International Conference on Multimedia. ACM, 2007:1009–1012.
- [15] Ramanathan P, Kalman M, Girod B. Rate-Distortion Optimized Interactive Light Field Streaming[J]. IEEE Transactions on Multimedia, 2007, 9(4):813–825.
- [16] Wu C, Tan Z, Wang Z, et al. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming[C]// ACM on Multimedia Systems Conference. ACM, 2017:193–198.
- [17] Schölkopf B, Platt J, Hofmann T. Graph-Based Visual Saliency[C]// International Conference on Neural Information Processing Systems. MIT Press, 2006:545–552.
- [18]<https://gpac.wp.imt.fr/2016/05/25/srdtuto/>
- [19]https://mp.weixin.qq.com/s/hpRiuWRW_ipt7IP2SjF1aA
- [20]<https://sites.google.com/site/duanmufanyi/publications>
- [21]Sitzmann V, Serrano A, Pavel A, et al. Saliency in VR: How Do People Explore Virtual Environments?[J]. IEEE Transactions on Visualization & Computer Graphics, 2018, PP(99):1–1.
- [22]罗传飞, 孔德辉, 刘翔凯, 等. 智慧家庭的VR全景视频业务实现[J]. 电信科学, 2017(10):185–193.
- [23]Gaddam V R, Riegler M, Eg R, et al. Tiling in Interactive Panoramic Video: Approaches and Evaluation[J]. IEEE Transactions on Multimedia, 2016, 18(9):1819–1831.
- [24]Bao Y, Wu H, Zhang T, et al. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos[C]// IEEE International Conference on Big Data. IEEE, 2017.
- [25]Bao Y, Zhang T, Pande A, et al. Motion-Prediction-Based Multicast for 360-Degree Video Transmissions[C]// IEEE International Conference on Sensing, Communication, and NETWORKING. IEEE, 2017.
- [26]Aladagli A D, Ekmekcioglu E, Jarnikov D, et al. Predicting head trajectories in 360° virtual reality videos[C]// International Conference on 3d Immersion. IEEE, 2018:1–6.
- [27]<https://mp.weixin.qq.com/s/4TmcBLXspzTeHUD0r61i0A>

第五章 全景视频 QoE 技术

5.1 主客观评价简介及主观评价的实施案例

沉浸式媒体应用与设备近年来的兴起，在一定程度上导致了 MPEG、3GPP、WebVR 以及其他相关领域标准化的推进。就目前的沉浸式媒体应用与设备而言，如何评价其体验质量并量化形成对比是十分必要的。就此，MPEG 已经出台了相应文件，大意上是要求沉浸式媒体添加与质量评估有关的额外输入以便进行对比和评价。

5.1.1 编码与数据流方案中的质量评估

如今，网络上每天产生的流媒体音频与视频数不胜数，占比也越来越大，因而编码与数据流在更多情况下是被绑定在一起的，且此趋势还会因 360 度视频等沉浸式媒体需要更多数据的情况下日益加深。

目前，沉浸式媒体内容的编码采用 HEVC 标准，因其可以降低通过 HTTP 协议传输动态自适应数据流时存储和带宽的要求，被认为是最先进最完善的编码方式。就此，诺基亚科技团队利用该标准在两种分辨率条件下储存同一全景视频，当向测试设备传输视频数据时，根据测试者的当前角度，部分 tile 通过高分辨率传输，剩余部分则利用低分辨率数据代替，类似方案在之前章节中也已提到过，其大致示意图如下所示。

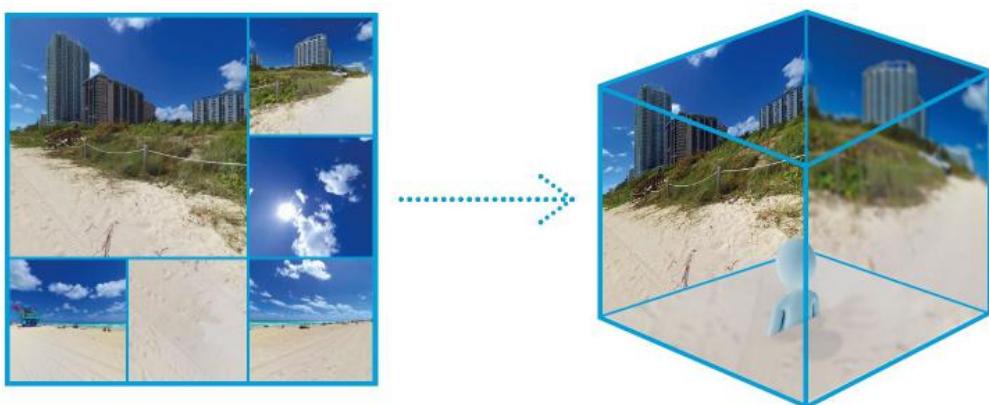


图 5.1 混合分辨率全景视频传输

同时为满足前述选择性传输的随机性，图像集中各 tile 均独立编码以便后续的解码，最终根据传输性能和压缩损失两方面的综合表现寻找到了一些列较优方案，并在这些可选方案中得到结论：相比于高分辨率传输所有视频内容，混合传输的方式可以降低 30%-40% 的比特率。此外，也有相关研究团队利用相同传输方式，对相同尺寸样式的图像块集在理想环境和实际环境下的脚本进行了详细的评估，包括比特率开销、带宽要求、峰值信噪比（PSNR）等多方面因素，部分结果如表 5.1 所示。实验结果与诺基亚团队所得结论相符合，并得到了更为全面的结论：对于实际的音频视频脚本，可以利用基于图像块的编码与数据流，节省至多 40% 的比特率；而对于理想脚本，在蜂窝网络中传输可降低近 80% 的比特率。

表 5.1 混合分辨率传输结果

Head Movements	Resolution	Tiling	BD-BR [%]		
			Tiles Monolithic	Tiles With Full Delivery Basic	
User 1	1920x960	3x2	30.538	-9.008	
User 1	1920x960	5x3	34.732	-35.427	
User 1	1920x960	6x4	38.680	-35.433	
User 1	1920x960	8x5	45.682	-35.360	
User 1	3840x1920	6x4	25.874	-38.982	
User 2	1920x960	3x2	30.779	-15.075	
User 2	1920x960	5x3	34.513	-28.976	
User 2	1920x960	6x4	38.501	-40.896	
User 2	1920x960	8x5	45.748	-29.970	
User 3	1920x960	3x2	31.042	-11.317	
User 3	1920x960	5x3	34.926	-31.786	
User 3	1920x960	6x4	38.884	-38.389	
User 3	1920x960	8x5	46.439	-32.282	

类似的 tile 集合也被用于提升媒体的交互性, 以及媒体服务质量指标计算的多个方面。法国 IMT Atlantique 团队于 2017 年 5 月发表的文章中提出了一种视角自适应 360 度视频传输的解决方案, 该方案需要服务器端提供同一内容视频的多种呈现方式, 也就是备有不同质量、不同分辨率的多种图像集, 而客户端设备根据用户视角向服务器请求合适的传输带宽, 此过程如下图所示。

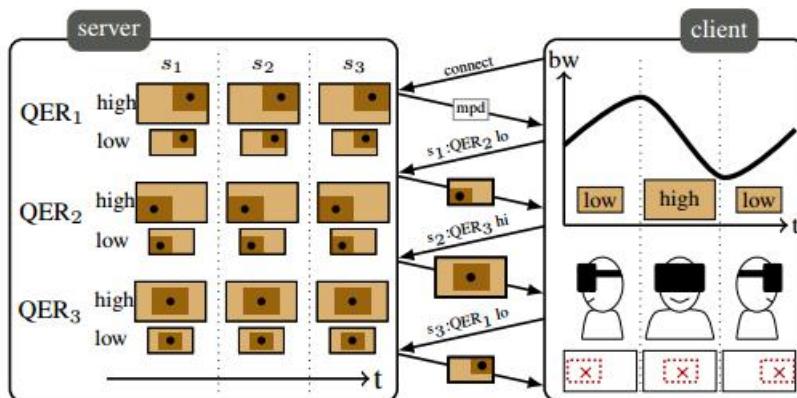


图 5.2 视角自适应的 360 度视频传输

同时, 该团队通过 PSNR 和图像质量评估算法 MS-SSIM 得到采用不同图像质量分布策略的数据流和图像投影方式对于最终效果的影响。此外, 也有文章提到在基于图像块的流传输时, 通过可变化的 IDR 帧的呈现方式可以减少传输拥塞的现象, 以提高传输质量。

德国弗劳恩霍夫应用研究促进协会近期的研究中提出了一项时空活跃性指标, 目的在于快速、简洁地计算出基于人们感兴趣区域 (ROI) 的视频传输方案。都柏林圣三一学院团队则基于 HTTP 标准和视频观看者的视角对动态自适应的数据流进行分割、整合, 产生良好的虚拟现实效果, 该团队也利用了 PSNR 及 SSIM 计算并验证了该传输方案相比已有方案, 更能切合用户的需求和期望。

除此之外, MPEG 文件中收录的与编译码、数据流相关的论文主要着重于比特率优化以及球面域失真优化的改良。比特率优化主要是指因全景视频的球面峰值信噪比 (S-PSNR) 与典型 PSNR 计算方式的不同而作出的基于比特率层面的优化, 使得传输字节合理分配至不同编码块时, 仍保持着可观的 S-PSNR。沉浸式视频相比于传统视频而言, 主要会在图像于二维/三维空间转换时产生失真, 而此类失真会使得数据传输时的率失真优化过程在一定条件下的作用微乎其微。对于该问题, 作出的改良是通过分析球面域失真对率失真优化的影响, 依此寻找出最优方案, 下图为实现过程。

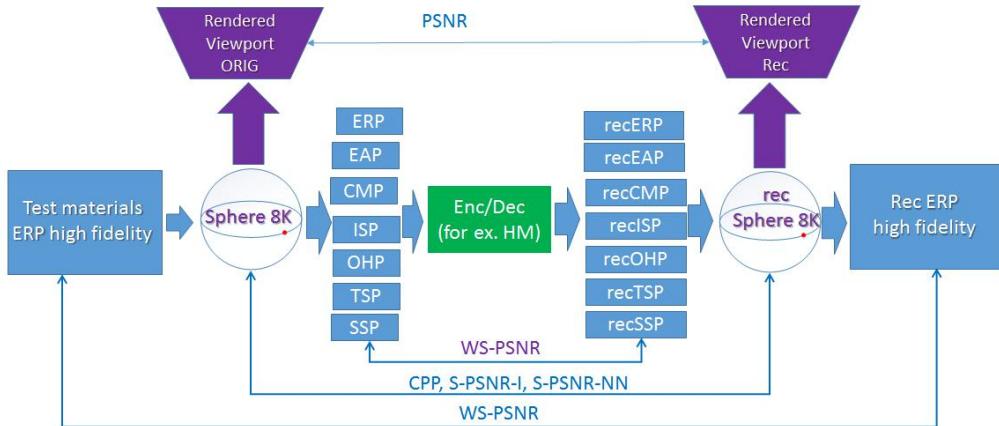


图 5.3 球面域率失真优化的改良过程

5.1.2 客观评价

上节众多方案大多采用的是客观质量评价，然而，许多度量沉浸式媒体质量的客观指标采用了衡量传统媒体的指标，或是略微修改指标的定义以满足 360 度图像或视频的特性，例如自适应性和视角认知度指标。但这样的做法目前存在的问题是，这类指标呈现出的媒体质量往往与人们的主观感受不匹配。就此，有研究人员认为，客观意义上的指标虽不能与人类体验达到完全一致，但仍会存在一定的规律。基于此，其团队利用全景图像样本，得到了大量的主观评分和客观结果，挖掘其中的统计规律，试图让单个客观指标具有类似于人的主观能动性。采用类似方法的基于视频样本的客观指标研究也已存在。当然，对于图像和视频两种表现形式而言，同种指标的计算也可能存在区别，例如图像与视频 PSNR 的计算，前者还需利用 SSIM、VIFP 等算法才能得到严谨的结果。

如上述对于特定指标的研究已有许多，然而目前还缺乏对包含多种编码传输方式，需计算多项指标的大范围样本的研究或评价。

5.1.3 主观评价

相比而言，对于沉浸式媒体主观指标的研究还不是很多。就目前来讲，360 度视频卡顿指标的计算已有研究，且得出的计算方式也可用于传统媒体，如电视、电脑、手机，其中电视与 VR 在相同条件下的卡顿程度对比如下图所示。同时该研究也对之后的卡顿研究提出了诸多建议，是相关领域中首次对头戴式设备如何预知卡顿的问题提出的见解。

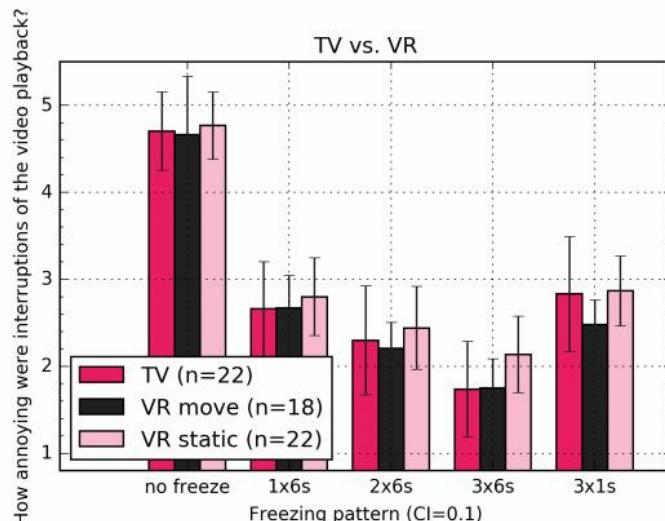


图 5.4 电视与 VR 的卡顿指标对比

另有一项关于 VR 视觉效果评价的研究通过主观测试以及秩相关系数检验（SRCC）来衡量 VR 内容质量以及各方向物体的一致性，并综合两方面表现提出了两种质量检测的主观指标：O-DMOS 和 V-DMOS。

此外，还有根据主观指标降低视频传输带宽的研究，而此项研究正是基于 DMOS 指标的基础上进行的，最终得到在 DMOS 值为 4.5 时，不同类型的视频传输平均可以降低 44% 的比特率，从结果上优化带宽的效果已比较可观。

实际上，在进行主观性能测试时，我们都假设测试者位于一个特定的环境中，例如一个常见的 VR 测试环境是让头戴设备的体验者坐在转椅上自由操作，或者规定其做一些特定的动作，然而即便如此，体验者们的感受仍可能迥乎不同，对于测试结果有着很大的影响。此外，当屏幕上放映内容，人们沉浸于其中时，相关测试设备也需统一。如果测试只关乎某项主观指标的问题，也只需做到统一就足够了，但是体验者在沉浸感、交互性以及其他方面的感受就会变差。

接下来，就将对主观评价的具体实施过程进行介绍。

5.1.4 主观评价的实施案例

由于人类的生物学因素和 HMD 的科技限制，360° 内容中每一时刻只有一小部分可以被测试主体观看和评估。观看全景的 VR 视频时，人们可以自由地观看 360° 的空间并且可以切换观看方向。这时产生的新的挑战就是如何确定来自同一测试主体或不同测试主体采用不同的评价方式得出的结果是可比的。换句话说，就是如何去确定测试主体是不是在同一时间对视频的同一个部分做出了评价。

已经有一些论文提出了对高分辨率或全景图像的评估方法。而接下来将介绍一种在数据流系统中主观评价全景视频的测试方法实施案例。

这一方法的基础来自现有的用于 2D 和 3D 视频主观质量评价的 ITU (International Telecommunications Union) 标准。但这一标准并不直接适用于 VR 视频内容，需要做一些扩展和调整。

主观视频评价实验应包括以下部分：1) 测试对象知情同意和个人数据收集；2) 预先筛选测试对象；3) 指示说明；4) 预测试问卷；5) 测试对象培训；6) 评分部分；7) 测试后问卷。

5.1.4.1 测试环境及对象

与通过平板显示器观看的传统视频不同，全景 VR 视频的典型观看环境需要 HMD 设备。要求测试对象在全方位空间中自由环顾四周，以评估场景的所有细节。为了避免在测试操作期间测试者被连接线缠绕，应优选无线 HMD 设备。与 2D 视频评价相比，全景视频评价任务具有更高的复杂性，不仅因为显示设备不同，还因为要求测试对象能自由环顾四周。因此，要尽量避免测试对象参与其他操作。评分方式最好采用口述的，比如向测试管理人员口述，由他将评分输入计算机用户界面 (UI)。让测试管理人员承担额外工作的作法的原因主要是担心使用自动化的 HMD UI 进行评分可能会导致测试对象的额外疲劳和/或输入错误的状况。为了便于移动，测试对象应该坐在旋转的椅子上或使用站立姿势。当对象头部方向移动时，视角改变并且当前视角的新内容显示给测试对象。

根据所执行测试的类型（初步测试或完整测试），可以组织专家测试对象或非专家测试对象的实验。专家测试对象应为在视频质量或相关领域工作的人员。受试者被筛选后，在初步测试中，建议最低专家数量为 5 人。在非专家测试中，测试对象不应在视频质量或相关领域工作，并且未在至少六个月内参与任何主观视频质量评估的活动。受试者被筛选后，完整

测试中，建议最少测试者数量为 15 人。如果测试结果中需要额外的统计可靠性，则建议最少数量为 28 个测试者（筛选后）。

由于测试活动可能产生健康方面的副作用，因此应进行入选前的健康认证。具有以下情况的测试对象应被排除在测试之外：怀孕，癫痫，其他神经系统疾病，强烈（过敏性）流感，强烈宿醉，特殊的短暂睡眠症，严重的一般眩晕症或戴着 HMD 的眩晕症。还应排除需要佩戴非常规 HMD 设备的测试对象。接下来，必须对招募的测试对象进行视力筛查，测试对象应该具有：1) 正常远视力和近视力（例如，使用 Snellen 图表或 E 图表测试）；2) 正常的色觉（例如，使用 Ishihara 板测试）；3) 正常的对比度感知；4) 正常的立体视觉（例如，使用随机立体点测试）。如果上述任何一项测试中显示有感知损伤，可以建议他/她不要进行评估实验。如果这一测试对象进行了实验，则应排除相关结果。此外，应记录受试者的瞳孔间距（IPD），因为 IPD 较短的受试者可能会出现更多的视觉不适。一些 HMD 具有补偿 IPD 差异的调整设置，可以使用这一设置进行调整。

5.1.4.2 评价方式

一种选定的评价全景视频的方式是单激励的 ACR-HR，这也是 ITU-T 推荐的方式。利用这种方法，视频测试序列（即已经应用了各种条件的源序列）将逐个地呈现给测试对象，并且在各个类别、等级上独立评分。每个测试序列的参考版本（即最高质量条件下的源序列）将作为其他测试视频序列显示给测试对象并且没有任何特殊标识（隐藏参考条件）。这就要求参考序列需要由视频专家评定为好或优秀。

测试过程中，主要向测试对象收集三种不同类型的评价结果：1) 视频质量分数；2) 视觉舒适度分数；3) 测试后的视觉疲劳程度。

视频质量分数的评价方式与普通视频评价测试一样，但如 5.1.4.1 中所述，分数收集是由特殊的口述报告完成的。视觉舒适度指的是观看立体图像时是否舒适的主观感受。当全景视频无法正确地捕获和/或显示时，测试者给出的结果将是一定程度上的不舒适。要注意的是，在独立的评价中，必须使用用于收集视频质量分数的相同测试序列来收集视觉舒适度分数。同时，一般需要在每个测试对象实验结束时填写测试后视觉疲劳度的问卷，以收集有关可能副作用的有价值信息。

对于前两个分数，使用 5 级量表（具有形容词类别判断）来收集结果。视觉质量：1-非常不好，2-不好，3-一般，4-好，5-非常好。视觉舒适度：1-非常不舒服，2-不舒服，3-中等舒适，4-舒适，5-非常舒适。尽管 ITU 建议在需要更大区分度的情况下使用超过五个级别（例如，对于低比特率序列）来收集结果，但更新的标准中还未要求增加级别数，因为 MOS 的准确性不会增加。如果需要更大的区分度，我们建议使用带小数的 5 级量表。

5.1.4.3 主观评价过程

视频质量评价实验的最后三部分（指示说明，评分和测试后问卷）构成了主观评价过程。如图 5.5 所示。推荐的实验总时间不应超过 1 小时，其中佩戴 HMD 设备的测试时间为 25-35 分钟。这个时间还包括 2-5 分钟的休息时间，在此期间测试对象可以移除 HMD 设备以放松。对于 180° 的视频内容，每个视频序列的建议长度为 20 秒，对于 360° 的视频内容，每个视频序列的推荐长度为 30 秒。

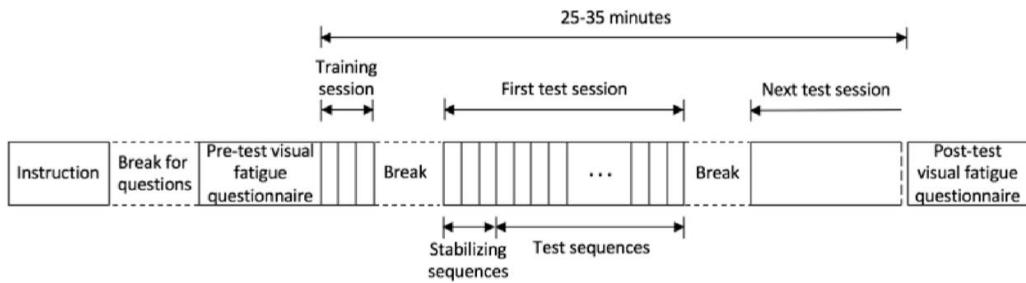


图 5.5 主观评价过程

指示说明

以下是从相关的主观评价标准规范中提取得到，并将其扩展为支持全景视频数据流评价的一组说明：

必须告知受试者可能由测试活动引起的负面影响。必须进一步告知他们可以随时拒绝或中止他们的测试且不会受到处罚，并且如果他们出现以下任何症状，应停止测试：眼睛疲劳，注意力不集中，一般疲劳，头痛，身体功能恶化，过度紧张，晕动症。要注意的是，如果这些症状在休息后消失，可以继续进行测试。

测试对象不会被告知测试中出现的视频损坏或损坏的位置类型。而以下内容应该告知测试对象：

- 正在测试的用例场景；
- 评估的目标；
- 整体评估程序；
- 每个测试部分中的任务。受试者必须环顾 360 度空间并探索（可选地，同时遵循特定的运动模式），避免停留在静止位置；
- 评估方法（他们将要看到什么，他们需要评估什么，例如，质量，绝对质量，视觉舒适度等方面差异）；
- 评分方式以及如何进行评分；
- 测试序列的数量和类型及其总持续时间；
- 不要将播放内容创建的程序、过程视为视频质量的事故（例如，SW 崩溃，偶尔闪烁，缝合伪像等）。这些不会影响评价的质量排名；
- 播放事故，例如长时间的加载和缓冲，它们不是测试评价的目标应当忽略；
- 不要让视频序列的审美程度和主题影响评分排名（例如，给定的视频包含无聊的内容）。

此外，在测试期间看到的最差质量不对应于评分量表上的最低主观等级。最后，要求测试对象填写测试前视觉疲劳度问卷，以引起他们对某些因素影响程度的注意：近视困难，夜视困难，双视，聚焦，在某个时刻的视觉质量，眼睛干燥或湿润，头后部疼痛，颈部僵硬，流感或宿醉或无法睡眠，眩晕（例如，观看 VR 视频时乘坐汽车船舶，乘坐游乐项目的情况）。

测试对象培训

对测试对象进行培训的目的是让受试者：1) 熟悉评分程序；2) 列出现有和待评估的全部范围（包括隐藏参考条件）的损伤类型，使他们的评分标准化；3) 鼓励提出有关实验的问题。这也是让受试者习惯于在全景环境中观看四周的机会（通过根据诸如下图中的特定模式环顾四周）。

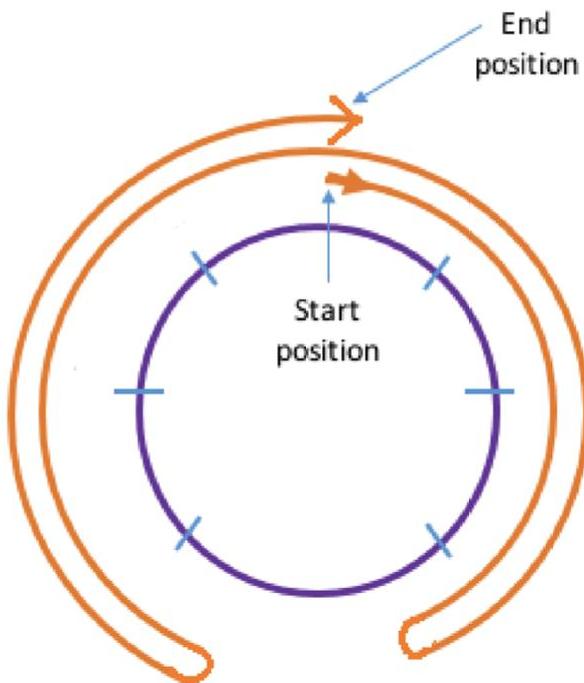


图 5.6 测试对象移动方式

我们建议使用至少五个视频序列进行培训，这些序列不会出现在实验中，但质量和持续时间与测试序列相当。这些序列的质量应涵盖实验中使用的所有测试序列的质量范围。在最终结果处理中不考虑培训测试中给出的评分。

测试方式

我们建议每次测试不应使用超过 20 个视频序列（包括稳定序列）。在每个测试开始之前，必须调整 HMD 设备以匹配测试对象的 IPD。可以通过观看测试静止图片来完成第一次调整。应记录下 HMD IPD 的值。

测试序列名称（例如“剪辑编号 1”）将使用扬声器播放和/或在 HMD 中可视地呈现，例如作为灰色矩形图案上的文本呈现。这样的矩形应固定在 HMD 屏幕的中间，对应测试对象在前一测试序列结束时的水平头部方向。这样能使测试对象在观看下一个测试序列之前无需返回到特定方向，能极大减少受试者的疲劳程度和眩晕症状。每一段测试序列播放结束后，提词屏幕上会呈现“请评价这一剪辑”的文字，持续大约 10 秒，在这期间测试对象应该用口述的方式给这一序列评分。

测试前及测试后的视觉疲劳度调查

该调查的目的是核对实验前后可能出现的副作用及其严重程度。对于全景视频的观看，已有的调查问卷应进行适当调整，并建议调查是否存在颈部僵硬或肩部疼痛，头部后部或前部疼痛或太阳穴疼痛，头晕，注意力不集中，恶心，困倦，眼睛刺痛，眼睛疲劳，聚焦困难，眼睛干燥或湿润，感觉眼睛朝不同方向看，双重视觉的症状，并调查在实验过程中受试者是否必须闭眼才能重建清晰的场景。

5.1.4.4 评价结果分析

该过程需要计算受试者们共同评价同一个测试序列的平均评价得分（MOS）。在本方法中，评分结果衡量了相同序列的两个版本（即参考和测试序列）之间的质量变化。因此，要在每个测试序列与其相应的参考序列之间计算 MOS 之差（DMOS）。报告的结果还应该包括

DMOS 的标准偏差和置信区间。利用这种方法可以从主观分数评价中去除参考序列的影响。

用于分析的一个基本数据集是测试对象的视角朝向数据。必须以足够的速率对空间中的头部朝向进行采样，以解决头部快速移动的问题。这种头部朝向由头部左右移动、上下移动和旋转的度数表示。该数据不仅用于验证同一测试主体是否在不同测试条件下评价 360° 视频序列的同一个部分，也用来确认所有测试主体的测试公平性。这里提出了两种方法来为计算头部运动模式的统计相似性创建输入数据：

- 1) 基于方向的头部运动模式 (OHM)：在每个测试序列内以特定间隔对测试对象头部方向（左右、上下、旋转）进行采样；
- 2) 基于视角的头部运动模式 (VHM)：在每个测试序列内以特定间隔记录被测试主体正在观看的视角。

由于全景视频内容具有比人类 FoV 宽得多的 FoV，因而如何公平地评价全景视频内容成为了一个难题。更具体地说，应该有一种方式来衡量：

- 1) 测试对象是否一直在观看所有视频序列（在同一时刻）的相同部分。（对象内相关性，即同一对象观看之间的相关性）
- 2) 多个测试对象是否一直在观看相同视频序列测试条件（在同一时刻）的相同部分。（对象之间的相关性，即不同对象观看之间的相关性）

相似环度量 (SRM) 则是一种目前可以在全景视频主观评价过程中使用的方法。其用于衡量不同测试用例之间观看模式的相似程度。它适用于单个测试对象和多个测试对象。该指标使用 OHM 输入方法收集的数据。基本思想是在 XY 图上绘制重叠曲线，其中 X 轴表示时间（视频序列持续时间），Y 轴表示头部方向。基于测试类型，每条曲线可以表示由单个测试对象观看的不同视频序列（相同的测试对象，不同的测试条件），或者由不同测试对象观看的相同的测试序列（相同的测试条件）。曲线（观看模式）重叠的程度代表了观看模式的相似性。

5.1.5 VQEG 的主观评价实例

VQEG 于 2018 年完成了一项主观评价调查，其采用了如下的 VR 体验调查问卷和模拟器副作用调查问卷：

How would you rate the picture quality? (circle the verbal option)

Bad	Poor	Fair	Good	Excellent
-----	------	------	------	-----------

How would you rate the responsiveness of the system? (circle the verbal option)

Bad	Poor	Fair	Good	Excellent
-----	------	------	------	-----------

How would you rate your ability to accomplish your task of loading the logs on the truck? (circle the verbal option)

Bad	Poor	Fair	Good	Excellent
-----	------	------	------	-----------

How would you rate the immersion of the experience? (circle the verbal option)

Bad	Poor	Fair	Good	Excellent
-----	------	------	------	-----------

How would you rate your overall experience? (circle the verbal option)

Bad	Poor	Fair	Good	Excellent
-----	------	------	------	-----------

图 5.7 VR 体验调查问卷

SIMULATOR SICKNESS QUESTIONNAIRE

Kennedy, Lane, Berbaum, & Lilienthal (1993)***

Instructions : Circle how much each symptom below is affecting you right now.

1. General discomfort	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
2. Fatigue	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
3. Headache	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
4. Eye strain	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
5. Difficulty focusing	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
6. Salivation increasing	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
7. Sweating	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
8. Nausea	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
9. Difficulty concentrating	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
10. « Fullness of the Head »	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
11. Blurred vision	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
12. Dizziness with eyes open	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
13. Dizziness with eyes closed	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
14. *Vertigo	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
15. **Stomach awareness	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>
16. Burping	<u>None</u>	<u>Slight</u>	<u>Moderate</u>	<u>Severe</u>

图 5.8 模拟器副作用调查问卷

以下是根据 17 名测试者得到的 5 项 VR 体验质量的 MOS 值：

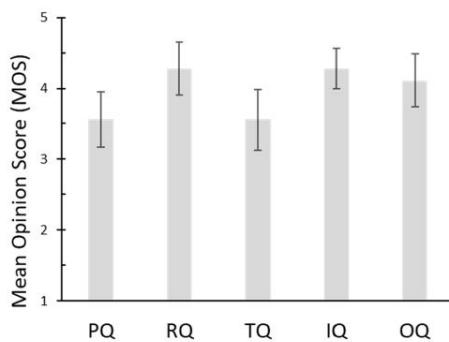


图 5.9 视频体验调查结果

除了基础调查外，VQEG 通过调整屏幕和操作杆配置，设置了 10 种不同的延迟情况。测试结果（图 5.10）显示，总体体验质量大致上是随着延迟的升高而降低的。但对于某些细化质量，如画面质量、任务完成度等，低延迟并不意味着数值的上升。

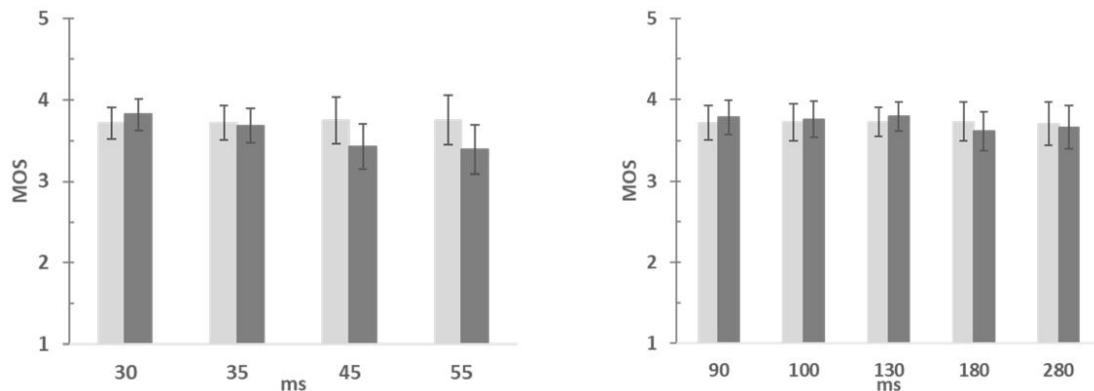


图 5.9 延迟调查-总体质量评分

5.2 典型的 QoE 要素

每个新技术的出现以及它的发展都会遇到很多问题，VR 在发展的道路上也会遇到许多瓶颈，这些瓶颈又会或直接或间接的影响用户的体验。就目前来说，不论是在 Oculus 还是 DayDream 平台，越来越多的消费者将使用 VR 设备的时间用在了观影上。这个数据也代表了用户的真实核心诉求，因为 VR 游戏从内容本身的体验上，还不足以满足大多数人们的要求，或许等到将来 VR 产业链和技术储备都成熟后，这种情况可以得到较大的完善。

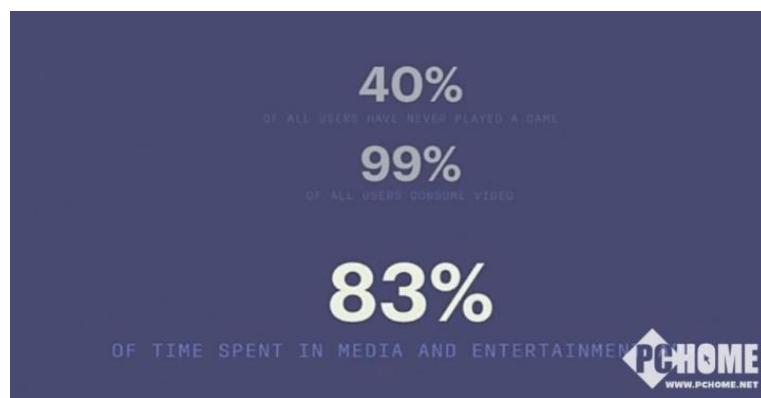


图 5.10 用户分布（来自 2018 Facebook8）

下面就介绍一些在 VR 发展过程中影响用户体验的要素和瓶颈。

5.2.1 视觉保真度

视觉保真度即是用户在使用设备观看内容时，所呈现的内容与原本的内容的相似程度。造成这种差别的根本原因就是数字设备呈现的画面是许多像素拼接而成的，这就必然会造成一定的失真情况。视觉保真度中有三个关键要素：纱窗效应，Mura(即亮度/颜色不均匀)和混叠。

纱窗效应

在许多初代的 VR 头显中，如 Rift 和 Vive 等，纱窗效应可能是最为明显的伪影。从技术来说，纱窗效应是“低填充系数”的结果，而“纱窗”的命名是因为这看起来就像是透过纱窗这样的精细网格来感知影像的。

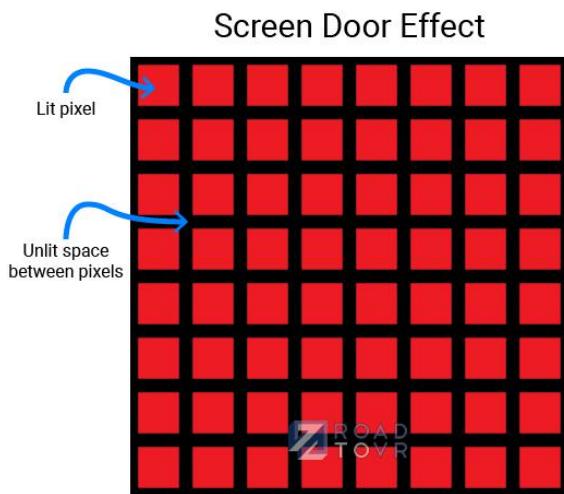


图 5.11 纱窗效应

像素是小而单独点亮的元素，它们通过阵列排放以建立显示。由于各种原因，像素有时难以紧密地打包在一起，而这导致它们之间的间隙没有点亮。显示屏的“填充系数”描述了实际点亮面积与非点亮面积之间的比例。在低填充系数的显示器上，使用者容易感知像素之间的非点亮空间，从而导致纱窗效应的出现。

此外，造成这种现象的原因还有 VR 眼镜的低分辨率，而且 VR 眼镜中是有放大镜的，这就更加容易导致纱窗效应的出现，就像本来是一张光滑的白纸，但通过放大镜来观察的话也会出现坑坑洼洼的现象。总的来说，纱窗效应是在像素不足的情况下，实时渲染引发的细线条舞动、高对比度边缘出现分离式闪烁的现象。

要想真正地消除纱窗效应，屏幕所呈现的内容一定需要有很高的分辨率。

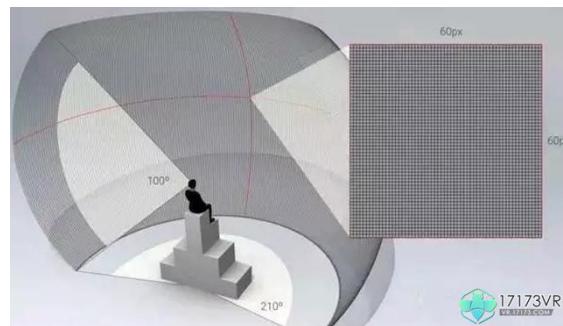


图 5.12 人的视场角

一般来说，每个人的视场角是水平 210 度左右，垂直 100 度左右，而要想看不到像素颗

粒感，视场角中的每一度要看到 60 个像素，即水平和垂直两个方向为 60*60 个像素（低于 60 个像素人类的视网膜就能分辨出像素粒）。所以水平的 210 度里就有 12600 个像素，垂直的 100 度里有 6000 个像素，理想情况下需要 12600*6000 的画面才能真正消除纱窗效应。但是即使 4K 的分辨率也才只有 4096*2160，在移动端都很难实现支持，距离理想情况有很大差距。因而结合目前沉浸式媒体的发展情况，现在需要的是硬件设备上的技术突破，而且手机端的计算处理能力也必须跟得上才有可能彻底消除纱窗效应的影响。

Mura (即亮度/颜色不均匀)

由于设备等原因，即使是由单一颜色值组成的一帧，从计算机输出到显示器时，也很难实现每个像素显示完全相同颜色。Mura 便是像素之间颜色和亮度不均匀的结果。如下图所示，显示器的所有像素都设定为一个颜色值，但由于设计和制作中的不完善，设备能力不足等原因，实际的颜色输出并不一致。

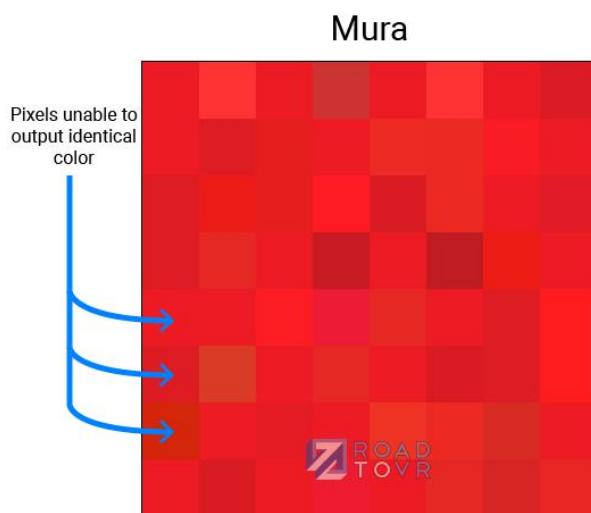


图 5.13 Mura (亮度/颜色不均匀)

不同显示屏的显示技术在此特性上有所差别，例如 LCD 在减少 Mura 方面往往表现得相当不错，而 OLED 表现就较差一些，需要仔细校准才能够产生良好的性能。

混叠

由于显示器由排列在网格中的方形像素（通常就是这样）组成，因此很容易显示与像素网格行对齐的水平线条和垂直线条。

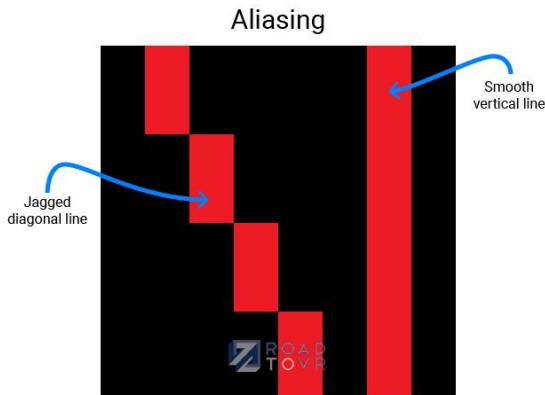


图 5.14 混叠

但在显示对角线或曲线时，就会产生问题：由于显示器只能用沿网格放置的方块来绘制

曲线，这样会导致非完全垂直或者非完全水平的边上出现锯齿。这意味着只有直线才能利用像素和像素网格的基本形状自然地显示出来。但是通过增加显示器的像素密度可以很直接地减少混叠的影响，显示器分辨率的提升能够更好地支持像素更精确地拟合出所需呈现的曲线。当然还可以通过抗锯齿的相关技术较少混叠的影响。

5.2.2 晕眩延迟

目前还有一个影响 VR 体验的大问题，就是晕眩感太强。一般来说，产生 VR 晕眩的根本原因是大脑对视觉和运动的认知不同步，即眼睛看到的内容(画面)与耳朵接收到的信息(位置)不匹配，导致脑负担加大，从而会产生晕眩感。VR 晕眩一般分为硬件和软件引起的晕眩。

硬件晕眩

VR 硬件中，主要有 5 种器件因其性能等原因会导致晕眩：GPU、感应器、显示屏、芯片成像透镜，以及瞳距和距离调整结构。一般来说，硬件晕眩是造成晕眩的主要原因。想要解决硬件晕眩问题，最简单的方法就是使用目前最好的硬件。但就目前来说，好的产品价格仍旧很贵，无法满足一般消费者的要求，所以现在需要做的，一方面是降低硬件的成本，另一方面还需要产业链的供给。

软件晕眩

VR 的软件晕眩一般有以下几大原因：

1) 游戏内容

很多 VR 游戏本身的内容就会产生晕眩。比如在坐“VR 过山车”时，视觉上正处于画面中的状态，在做剧烈的高速运动，但是前庭系统却并没有感知到运动状态，这时就会导致头晕，产生晕眩感。

2) 画面与现实世界的差异

VR 显示器给出的画面变形严重、视野小，这些都与现实世界存在差异，时间长了就会感觉头晕。

3) 画面滞后于动作

VR 中的 MTP 延迟造成时间上的不同步，当人转动视角或是移动的时候，画面呈现的速度会有些慢跟不上头部的运动。在 VR 这样全视角的屏幕中，延迟也是造成晕眩的关键因素。而且现在许多的 VR 设备均是基于 Android 系统，但传统的 Android 系统都是采用了双显示缓冲或者三显示缓冲，这个机制让 VR 应用无法知道制定的图像什么时候能够显示在头盔屏幕上，这就会造成更大的延迟。

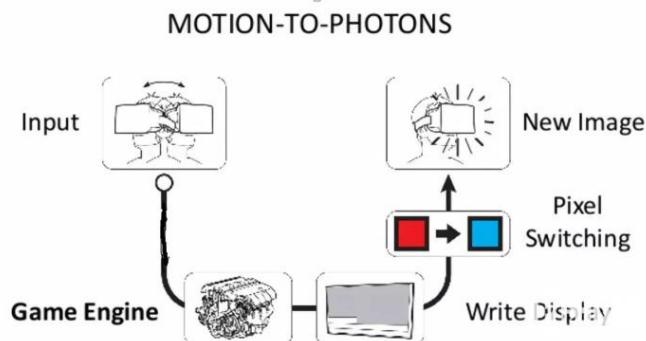


图 5.15 MTP 延时

4) 瞳距不一

由于每个人的瞳距不一，对某些人来说，人眼瞳孔中心、透镜中心、画面中心三点并非

一线，从而出现重影现象，看久了也会非常容易头晕。

5) 景深不同步

景深不同步，也是眩晕的原因之一。比如说，在你的面前，有一张桌子，在桌子上，近处放了一个杯子，远处放了一个玩偶。你看着近处的杯子，按理来说远处的玩偶应该模糊不清，但是现在，远处的玩偶也看的非常清晰。这是因为 VR 设备对现实的模拟还不够真实，无法真正地欺骗到大脑，受到困扰的大脑不堪重负，才会造成眩晕。

所以，如何解决 VR 眩晕问题呢？下面将介绍一些能够解决眩晕问题的方法。

低延迟技术

这里所谓延迟指的是从头部转动到画面转动的延迟。画面延迟在很大程度上又取决于显示屏的刷新率。目前世界上最先进的虚拟现实设备刷新率在 75Hz 左右。研究表明，头部运动和视野的延迟不能超过 20ms，不然就会出现眩晕。20ms 的延迟时间对于 VR 头显而言是一个非常大的挑战。首先设备需要足够精确的办法来测定头部转动的速度、角度和距离，这可以使用惯性陀螺仪（反应灵敏但是精度差）或者光学方法来实现。然后计算机需要及时渲染出画面，显示器也需要及时地显示出画面，这一切都需要在 20ms 以内完成。相应的，如果每一帧显示的时间距离上一帧超过 20ms，那么人眼同样也会感到延迟。所以，VR 头显的画面刷新率应该超过 50FPS，目前来说 60FPS 是一个基准。但是要想做到更好的效果，这个刷新率还应该接着往上提高，比如目前 Oculus Rift CV1 和 HTC Vive 采用了 90Hz 刷新率，而 Sony Project Morpheus 采用的是 120Hz 刷新率。

一般说来，延迟产生的原因很多，在之前也提到过，首先从头部转动到传感器读到数据，然后数据需要经由单片机通过 USB 线传输到电脑，在硬件上传输完成后，就是软件算法处理过程，在将模拟信号转换成数字信号之后，数据中存在大量的噪声和漂移，于是需要复杂的数字信号处理方法将这些噪声和漂移过滤掉，接下来是运用 Time-warp 算法渲染场景，在渲染完后还需要做反畸变和反色散等处理，最后还需要的时间延迟就是传输图像到显示器的时间，不过现在的 OLED 技术已经将这个时间较少到了微秒级别。延迟中还涉及到 CPU 的性能、USB 丢包等问题。

目前看来，从 VR 硬件方面降低延迟是改善眩晕最好的方法，所有需要的处理时间不断压缩便能够降低画面的延迟。其次就像之前的 3D 眩晕一样，使用者经过一段时间的适应，才能使 VR 成为真正意义上的虚拟现实产物。

添加虚拟参考物

普杜大学计算机图形技术学院的研究人员于近年发现，只要在 VR 场景中添加一个虚拟的鼻子，就能在一定程度上解决头晕等问题。研究人员在各种虚拟场景中对 41 名参与者进行了测试，一部分人会有虚拟鼻子，一部分没有。结果发现，有鼻子的人都能保持更长时间的清醒。研究人员称，之所以能产生治晕的效果，可能是因为人需要一个固定的视觉参照物，所以就算在 VR 中加入一个汽车仪表盘，也可能会产生相同的效果。

电前庭刺激

目前国外正有一家名叫 vMocion 的公司，打算利用梅奥医学中心的航空航天医学和前庭研究实验室花费 10 多年时间研究的技术去解决这个问题。这项技术名为电前庭刺激 (GVS)，将电极放在策略性位置（每只耳朵后要放置两个电极，一个在前部，一个在颈背），追踪用户内耳的感知运动，并将视野范围的运动触发成 GVS 同步指令，刺激产生三维运动。如果行得通的话，它可以让用户完全沉浸在当前的环境中，真正感觉到自己驾驶的宇宙飞船在俯冲或转弯。

调节镜片之间的距离

在最原始的 HMD 中其实并没有考虑到这一点，考虑更多的只是怎样适配不同程度近视的用户（有些 HMD 甚至没有考虑这一点），三星首先采用了滑轮调节镜片的设计，可以自由调节两个镜片之间的距离。另外，也有方案显示可以通过蓝牙控制器等调节画面的中心点。从而保证画面中心、镜片中心、人眼中心三点一线。避免重影，避免晕眩。

光场摄影

一个光场快照可以在图片获取后对照片进行聚焦、曝光、甚至调整景深等操作。它不仅仅记录落在每个感光单元内所有光线的总和，光场相机还旨在测定每个进入光线的强度和方向。有了这些信息，就可以生成不只是一个，而是每一个在那一刻进入相机视野的可能的图像。例如，摄影师常常会调整相机的镜头，以便对面部进行聚焦，刻意模糊背景。也有人想要得到模糊的面部，背景要十分清晰。有了光场摄影，同一张照片，就可以获得任何效果。目前在这方面做得最好的 Magic Leap。Magic Leap 做的不是在显示屏上显示画面，而是直接把整个数字光场投射到使用者的视网膜上，从而可以让使用者可以根据人眼的聚焦习惯自由地选择聚焦的位置，以准确地虚实结合模拟人眼的视觉效果，而完全不会涉及到刷新率和分辨率等问题。

5.2.3 视角

传统设备中的视角处理方式包括镜头控制加角色控制，比如固定摄像机位置，游戏中的角色自由移动，或者摄像机随着游戏中角色的移动而移动，还有就是当观看全景视频时，镜头是跟着头部转动而移动的，也就是每次看到的都是整个球面中的一部分即视角区域，而且当在 VR 游戏中时，游戏设计师无法控制玩家的视角，任何类型的 VR 游戏都是主视角游戏，相机的行为控制是游戏是否会带来晕眩感的关键因素。

5.3 球面指标测度

目前，VR 市场上的产品质量参差不齐，需要有统一的标准来进行评价和判断。第 120 届 MPEG 会议上，ITU-T SG12 Q13/12 文档中概述了新的工作项目（虚拟现实的体验质量），包括应用场景和相关的 QoE 因素。其工作的目标是开发主观测试方法和质量模型，并且实际上已经有相关组织开始为 360 度 HMD 视频和 VR 游戏的主观评估提出一些新的建议。目前的基准提供了 VR 硬件和软件，VR 使用案例，影响因素分类，以及 VR QoE 指标。在 ITU-T SG12 全体会议上，华为公司提出了 QoE for VR (Quality of Experience for Virtual Reality) 工作立项提案，该工作项目将输出关于 VR 业务质量领域的 Recommendations 文档，向业界提供关于 QoE 因素、QoE/QoS 方面需求、主观测试方法和客观质量评价模型等方面的建议。

Facebook 也一直致力于推动 360 度技术的发展，而且也开拓创新了一系列的新概念，比如说有偏移立方体贴图 (offset cubemaps)，动态流式传输 (dynamic streaming) 和基于内容的流式传输 (content-dependent streaming)。每一项新技术都对 360 度技术有重要的作用，同样重要的是每一项新技术与原先方法比较时的体验质量，即 QoE 的提高，最好能够提供一致的标准，并且能够对以后新出现的技术也作出衡量。

然而，目前 VR 行业还没有统一认可的 360 度内容评价标准，而传统的视频评价体系无法对 360 度视频的相关属性做出评价，如沉浸感和观影控制等。360 度视频每一帧都压缩了一个全方向的球形场景，如果要利用传统的评价体系，还需要将 360 球形帧还原成矩形帧，

这种翘曲操作会使得评价结果变得相当不可靠。而且，用户在 360 度视频播放期间可以控制任意时刻的观影方向，这也就意味着最终的视图质量取决于用户视场的帧区域，而不是整体的球形帧。

5.3.1 Facebook: SSIM360 和 360QVM

SSIM360

首先简单介绍一下传统非 360 度 SSIM。结构相似性指标(Structural Similarity Index, SSIM)是广泛应用于图像和视频编码的 QA 指标。评估过程为先输入两幅图像：参考图像（如原始内容）和变化图像（如编码内容），然后输出是 0 和 1 之间的分数。分数代表两幅图像的结构相似性，1 代表图像相同，因此质量保存完好，0 意味着两者完全不同，意味着严重的结构变形。该评估过程是逐帧完成的，而且仅在两个输入视频的长度和帧速率完全相同时才起作用。

跟大部分的图像处理算法类似，SSIM 不会一次性测量整幅图像。相反，它从两个输入图像中采样较小的区域（类似于图像压缩中的宏块）并比较样本。在最初的论文中，作者提出了一个用于这一采样的 11×11 高斯内核，而 ffmpeg 的 vf_ssim 滤波器使用了一个统一的 8×8 方块。从每个样本中获取一个 SSIM 值。为了获取每幅图像的 SSIM，计算每个样本的 SSIM 平均数。为了获取整个视频的 SSIM，进一步平均计算所有帧的每幅图像 SSIM。

SSIM 中的采样区域是静态的（即相同的大小，相同的形状），并且在最终的平均计算中具有相同的权重。例如在上面的 ffmpeg 实现中，不管位置如何，图像中每一个 8×8 区域对最终的 SSIM 分数都具有同样的影响力。但 360 度媒介并非如此。前面提到的翘曲问题可能会使得同样的图像或视频得到不同的 SSIM。

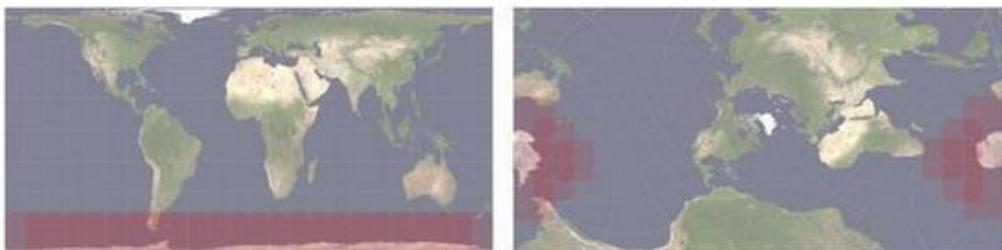


图 5.16 不同方向的球体展开

比较上面的两张世界地图。左边地图的编码器会降低画面下部的质量（南极洲）；而右边地图只是左边编码地图的重定向版本。当渲染至 3D 球体时，这两张地图应该看起来完全一样（除了方向），所以它们从 360 度内容 QA 中得到的分数也应该是相同。但是，如果我们使用 SSIM 作为我们的 QA 标准，则左边地图分数会较低。原因是由于等量矩形中靠近垂直中心的翘曲较少，所以右边地图“糟糕质量”部分（南极洲）所占的比例较小。在 SSIM 的 8×8 取样方块中（在两个地图上呈现为红色放宽），左边地图中南极洲占据的区域被采样 23 次，而右边地图只有 12 次，所以这导致了不同的平均 SSIM 分数。

一种防止翘曲影响的方法是，直接对 360 度球体中的渲染视图进行取样，而非采用等量矩形的渲染视图。我们从 360 度球体中的所有可能视角方向获取无限数量的方形快照。这种样本是平坦图像，同时进行了加权平均，所以翘曲问题将不复存在，而且 SSIM 将是一个有效的 QA。为了防止翘曲，在馈送至非 360 SSIM 之前先获取平面图像的快照 (V_1, \dots, V_n)。

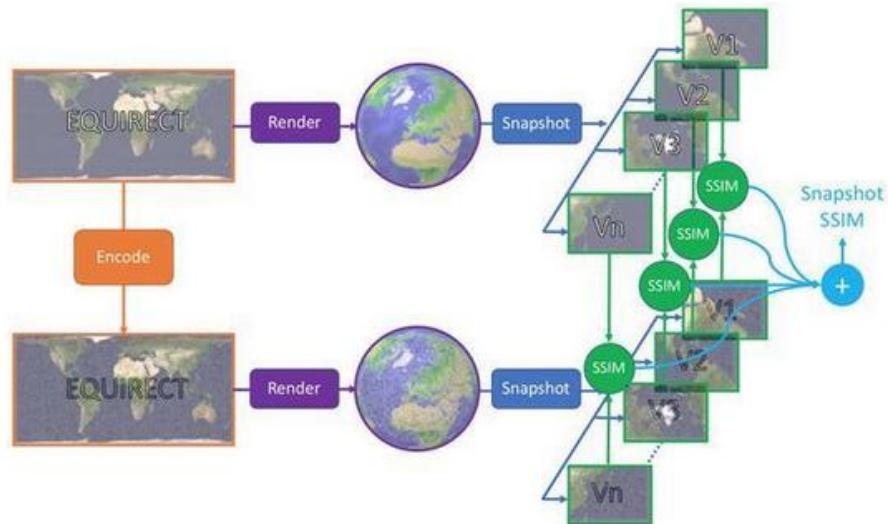


图 5.17 快照方式解决翘曲问题

最终的“Snapshot SSIM”分数将通过汇总所有的每快照 SSIM 分数进行计算。虽然可靠，但这种方法的计算复杂性使其成为不可行的 360 度图像 QA 解决方案。从给定视图中获取单个快照并不是一件简单的事情，更不用说获取无限数量的快照并且逐个运行 SSIM。但是，我们可以使用汇总的快照分数作为验证我们新 SSIM360 标准的基础事实。下面介绍 Facebook 提出的 SSIM360。

SSIM360 解决翘曲问题的方法是，在计算平均值时为每的样本 SSIM 得分加权。加权取决于采样区域在影像中的拉伸程度：拉伸越多，加权越小。这可以通过“样本覆盖的渲染球体比例”和“样本覆盖的帧比例”的比例进行计算。每个样本对最终分数的贡献不同，所以能有效地消除翘曲影响。

为了验证 SSIM360 的结果，将其与 Snapshot SSIM 进行比较。针对不同的纹理和动态特征的 360 视频，以及不同的编解码器，缩放和质量保存目标进行了验证实验。然后通过 1) SSIM，2) SSIM360 和 3) Snapshot SSIM 评估这些降级测试用例。

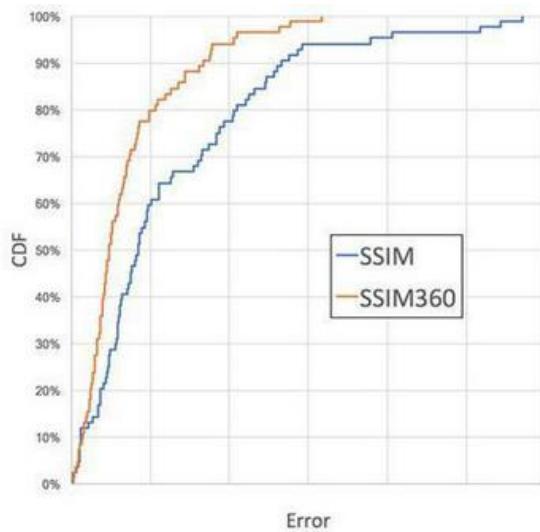


图 5.18 不同 SSIM 性能比较

上图中是 SSIM360 和 SSIM 的累积分布函数 (CDF)，可以看到 SSIM360 几乎能在所有的百分比中把错误减少约 50%。错误定义为 Snapshot SSIM 分数与每个 QA (SSIM360, SSIM) 输出分数之间的绝对差异。从计算角度而言，由于 SSIM360 只是通过可预先计算的权重图来替换每样本 SSIM 聚合中的统一加权，因此 SSIM360 与传统 SSIM 一样有效。

在 360 度视频中还涉及到视场问题，SSIM360 是在不确定用户注视点的情况下评估编码级别的质量。但在播放质量评估中，我们必须考虑到视图方向。在任何特定时间下，只有大约 15% 的 360 度场景会保留在用户视场中。这意味着通过 SSIM360 计算的整体球形帧质量将不再具有代表性。当在内容交付框架中采用基于视图的优化时，情况尤其如此。基于视图的优化技术（如偏移投影，基于显着性的编码，以及基于内容的流式传输）基本上将位分配（相当于大多数情况下的像素分配）偏向视频中感知更重要的区域。它们并没有提高整个画面的质量，而是优化用户最有可能注视的区域。而 SSIM360 由于侧重于整个画面无法捕捉到这种优化。

Facebook 通过两个映射解决了视场问题：一个是从视图方向到像素密度，另一个则是从像素密度到应用于 SSIM360 得分的比例系数。第一个映射考虑了编码中使用的投影，以及播放期间的视图方向和视场，并且询问每个时间点内有多少像素停留在视场之中。换句话说，FoV (Field of View) 中有多少像素，通过已知的像素密度图，可以通过封闭型几何公式来有效地计算答案。

第二个映射通过引入反映像素密度变化的比例系数来调整 SSIM360 的质量分数。在非 360 度 QA 中，会使用类似的技术来评估缩小尺寸所导致的质量下降（即将图像“缩小”到更小的宽度/高度）。由于 SSIM 假定参考图像和变化图像具有相同的尺寸，因此缩小后的图像必须放大至原始尺寸，并重新采样以评估质量下降情况。不可以忽略这所需要的计算工作，特别是评估转码过程。转码过程会产生数十种具有不同维度的编码格式，而所有这些都需要在单独的 QA 过程中重新采样，如下图所示。

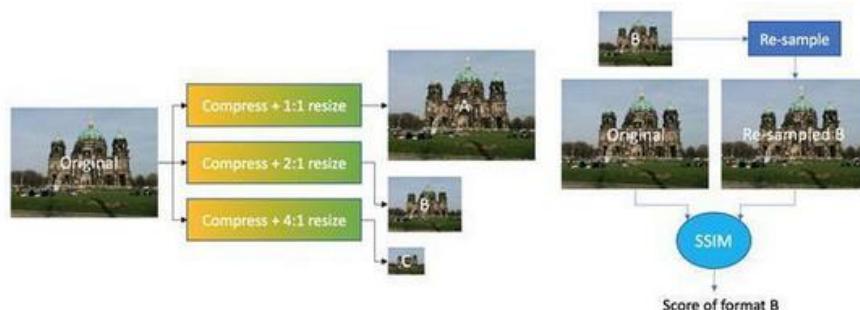


图 5.19 SSIM 过程需对不同大小的图像重新采样

要避免这种计算开销，我们可以将转码过程的 QA 解耦为两个步骤：压缩的 QA；调整大小的 QA。如下所示，SSIM 在压缩版本上执行，同时不调整大小。然后通过将惩罚因子应用于压缩版本来近似计算每个调整大小的格式的单独质量分数。调整大小的比例是决定惩罚因子的一个因素：调整大小的版本越小，损失的结构细节越多，所以惩罚因子越高。通过确定惩罚因子来取代重新取样更为有效，并且能够减少我们运行 SSIM 的次数。

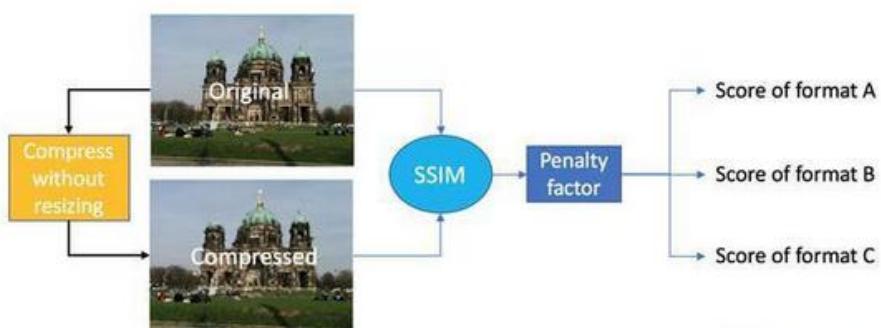


图 5.20 惩罚因子

在 360 度视频中，分辨率的变化不是来自调整大小，而是来自视场内像素密度的变化。随着用户注视点朝更高/更低像素密度的区域移动，观看内容（即视场内的内容）的分辨率将随之增加/减少。比例系数与惩罚因子不同，因为它可以减少或增加 SSIM360 得分。对于基于视图的投影，视场内的分辨率会发生变化，就好像视频分辨率通过“视场内的像素密度”与“整个帧的像素密度”之间的比率进行升级或降级一样。

360VQM

通过将以上的技巧应用到定制的 QA (quality assessment) 工作流程中，就可以解决 360 度视频会话中 QA 的翘曲和视场问题。SSIM360 取代 SSIM 来处理编码中的扭曲，而惩罚因子则由比例因子所取代（因视图改变引起的像素密度变而衍生出来）。这一 QA 工作流程的结果称为 360VQM：360 度视频质量指标。它可以准确高效地捕捉编码过程中的质量变化并在播放过程中将分辨率变化的影响体现在最终分数上。

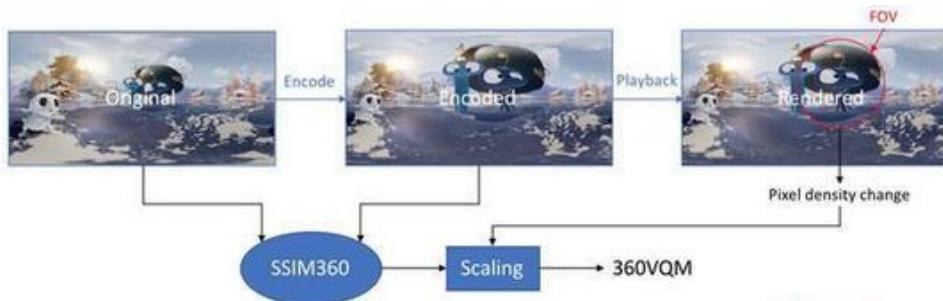


图 5.21 360VQM

5.3.2 VQEG/ITU 中的 3DTV

VQEG 是视频质量专家组 (Video Quality Experts Group) 的简称，该组织专门研究视频评价，其中的 Immersive Media Group 负责沉浸式媒体的质量评价，包括虚拟现实，增强现实，立体 3DTV，多视图等。

目前来说，该组织正在制作一组独特的视频序列来进行实验 (GroTruQoE 数据集)，第一步是使用配对比较方法进行大规模实验，因为序列比较简单，受试者可以轻松地提供对总体偏好的判断。相应的测试计划是“为 3D 视频质量评估中的评估方法建立 3D 体验质量的基础真相” (GroTruQoE3D1)，第二步将使用配对比较测试的结果作为 groundtruth 数据库，以研究哪种更具时效性的主观测试方法可用于预测配对比较测试的结果。

5.3.3 Visbit 360 基准视频

VR 行业有很多关于 4K 与 8K，传统渲染与注视点渲染等话题的比较讨论。开发者可能想知道以 8K 质量制作的内容是否会对 360 度 VR 视频用户体验产生任何影响。VR360 度视频的最终清晰度受到源视频质量、压缩、渲染管道、屏幕分辨率、屏幕像素结构、VR 镜头质量，甚至是眼睛锐度等多种因素的影响。更复杂的是，这一切无法简单地叠加。这涉及大量的重新采样，混叠和插值。最后，即使没有上述的复杂问题，当每个人都用他们自己的视频进行比较时，仍然很难轻松分辨出差异，因为其质量和分辨率都有所不同。

为此，Visbit 推出了基准视频 (VB2018VR)，可以在各种播放器中轻松进行测试，并且判断其质量差异。下面是 Visbit Benchmark 360 VR Video (VB2018VR) 的截图，它可以用来自测试、评估和选择最佳的播放解决方案，也可以用来检查后期制作流程，判断管道中是否存在任何质量丢失。

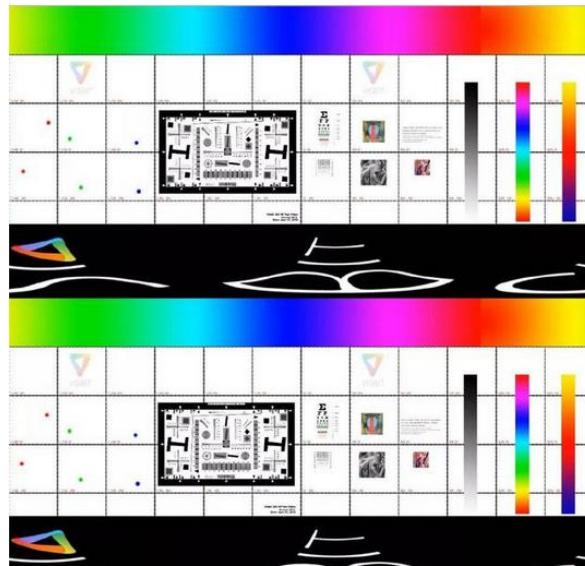


图 5.22 VB2018VR V0.5 的截图 (7680 x 7680 TB, YUV444p)

这个基准视频长约 10 秒，采用 7680×7680 分辨率（立体上下），色域为 YUV444p，速率是 30 FPS，采用 H.264 编码（如需 HEVC 版本，可联系 Visbit）。视频包括三个可视区域，专门用于测试细节和可读性、动态范围和帧速率。

测试的时候，只需要将 8K 基准视频上传至任意平台或播放器，即可进行比较：(1)肉眼比较；(2)截图比较。其中有关于颜色矫正的注意事项：(1)为补偿 VR 透镜的颜色失真，有的播放器（如 Visbit 8K VR Player）会校正颜色。色彩校正可以改善 VR 中的观影体验。但在屏幕截图中，你可能会看到锐利边缘出现色溢，从而令细节变得模糊。(2)如有需要，你可以获取使用非颜色校正版本以进行基准测试。(3)否则，你应该仅比较视场中心，因为这里的颜色校正程度最小。

如果你瞄准的是最高清晰度，以下便是截至 2018 年 6 月的映维网推荐的测试硬件：Pico Neo，HTC Vive Focus 或者搭载 Galaxy S9 的 Gear VR。在观影基准视频时，以下是可能用到的各种质量测试。

视场测试

对于这个基准视频，你可以轻松看到每个头显的视场。需要注意的是，VB2018VR 是以度来表示视场角。打开头显并播放视频后，你就可以读取水平和垂直视场。例如，Gear VR + S9 上的视场大约是 70 度。

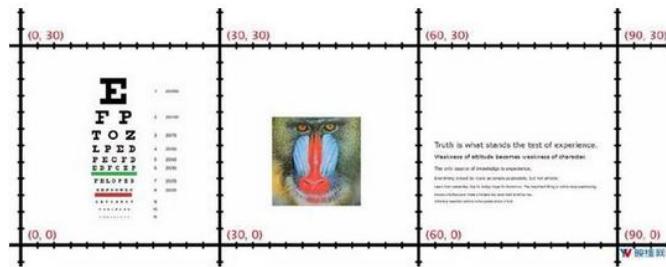


图 5.23 VB2018VR 中的视场角

细节与可读性测试：面部与文本

面部和文本可能是最重要的两种对象。通常而言，人类对面部和文本更加敏感。下面将纳入著名的 Lena，Barbara 和 Baboon 图，以及包含不同字体大小的文本，如下图所示。

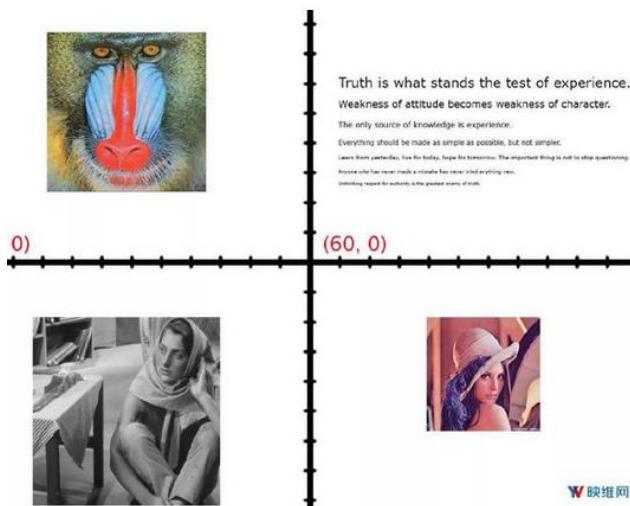


图 5.24 面部与文本

要留意的内容：能快速阅读哪一行文字？是否能清楚地看到面部的细节，如眼睛，嘴唇和鼻子？是否能看到女人衣服或动物皮毛上的纹理？

细节与可读性测试：视力表

VB2018VR 纳入了两张视力表。你最低能看到哪一行呢？要注意的是，左边的视力表本身有点模糊，VB2018VR 将在之后进行优化。

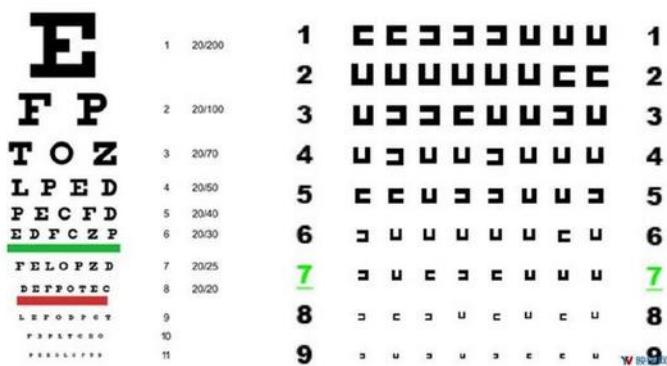


图 5.25 分辨率测试标版

下图则包含了典型的 ISO-12233 分辨率图表。图标中的锯齿十分明显。你可以容忍多少锯齿呢？这个问题有点主观。回答这个问题前你应该回到面部与文本的测试，然后再来看看是否能接受其锯齿程度。

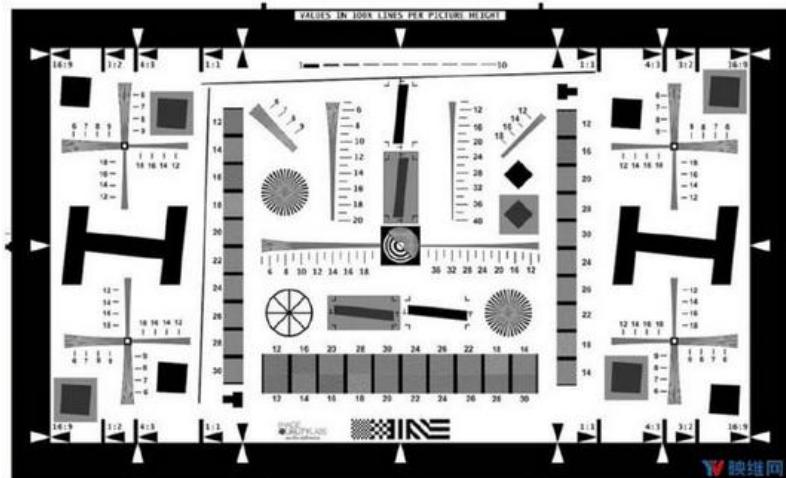


图 5.26 ISO 分辨率测试标版

动态范围测试

如果播放器没有很好地处理色域或动态范围，你将看到渐变颜色出现带状效果。

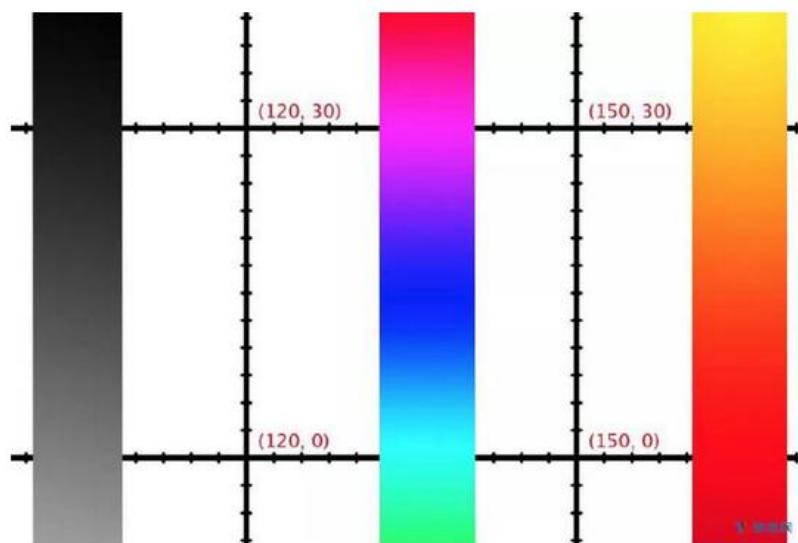


图 5.27 用于动态范围测试的色彩渐变

帧率测试

VB2018VR 同时包括以不同速度和 3D 路径移动的多个点。另外，视频最上方和最下方正在快速旋转。你可以依次判断运动的流畅度。

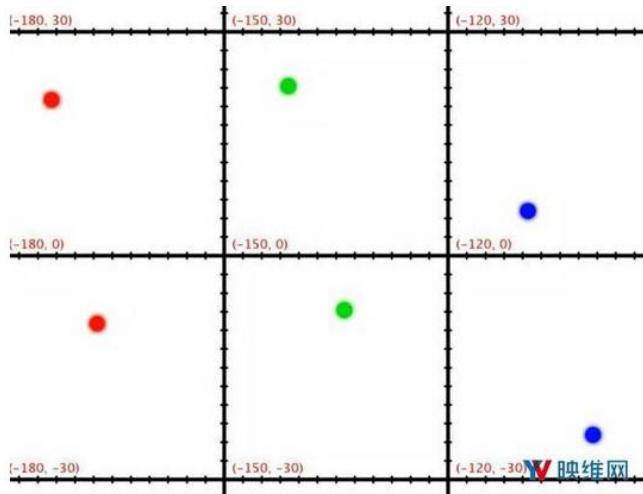


图 5.28 帧率测试的移动点

文件对比测试

VB2018VR 提供了一系列的测试视频，内容相同，只是分辨率不同。你可以自由下载并用任何视频播放器进行测试，然后分享比较结果：

1. visbit-8k-stereo: YUV444P, 7680 x 7680 @ 30fps
2. visbit-6k-stereo: YUV444P, 5760 x 5760 @ 30fps
3. visbit-6k-stereo-yuv420: YUV420P, 5760 x 5760 @ 30fps
4. visbit-4k-stereo: YUV444P, 3840 x 3840 @ 30fps
5. visbit-4k-stereo-yuv420: YUV420P, 3840 x 3840 @ 30fps
6. visbit-4k-squeeze-stereo: YUV444P, 3840 x 1920 @ 30fps
7. visbit-4k-squeeze-stereo-yuv420: YUV420P, 3840 x 1920 @ 30fps

5.3.4 游戏体验质量建模

在消费级沉浸式媒体服务中，游戏是占很大比例的一项内容。相比于 2D 屏幕，在虚拟全景世界中进行游戏体验是极为真实刺激的。从目前火热的游戏直播的角度而言，Twitch TV 是其中相当成功的一个平台，据统计，2015 年 Twitch TV 每月的访问者有近 1 亿。游戏直播是指游戏主播向平台传输游戏画面，并通过直播平台向广大观众展示自己的游戏过程。与这一概念类似，用户终端无需任何高端处理器和显卡，只需基本视频解压能力就可以直接进行游戏的方式被称为“云游戏”，而在沉浸式媒体领域，“云游戏”的最大瓶颈就在于延迟。

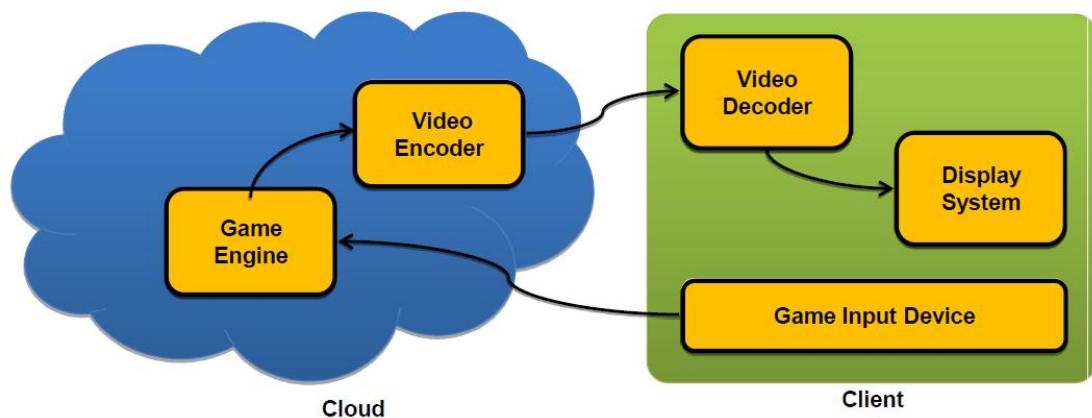


图 5.29 “云游戏”

随着类似上述的各类游戏方式的诞生，游戏体验质量的评估也有待完善。目前，ITU-T的Q.13/SG 12、Q7/SG 12项目都涉及到游戏应用的QoE。VQEG-IMG工作组于2018年根据多编码方式、多分辨率、多比特率版本的16个游戏视频对游戏QoE完成了一项研究。该研究利用多种特征如平均差（MAD, Mean Average Difference）、空间信息（SI, Spatial Information）、时间信息（Temporal Information）进行游戏体验质量的建模，其大致建模框架如下图所示：

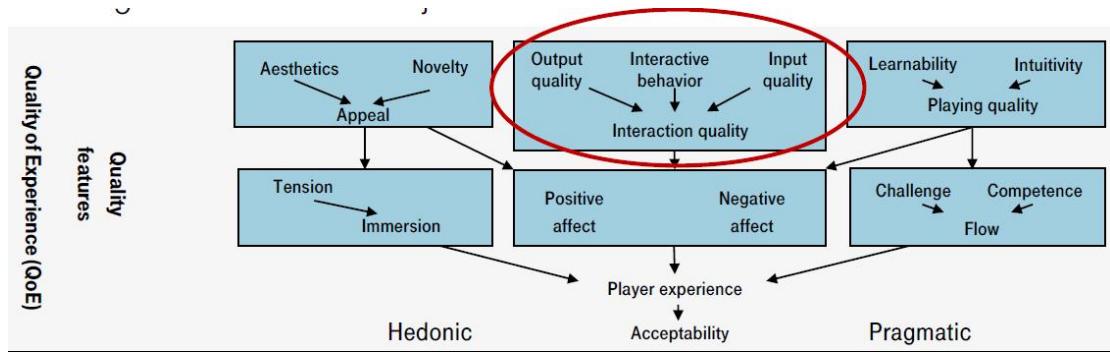


图 5.30 游戏体验质量建模框架

多重因素建模的QoE分数与主观测试的MOS进行拟合，最终得到以下等式：

$$MOS = 1.102 + 0.59 * PosAffect + 0.24 * Reactiveness + 0.25 * VideoQuality \quad (5.1)$$

其中， $Reactiveness = \exp(0.84 + 4.43/\text{帧率})$ 。此模型是一种无参考质量评价算法，同时未涉及比特率模型，游戏中的正面影响和输出流是两种主要的分数贡献因素。

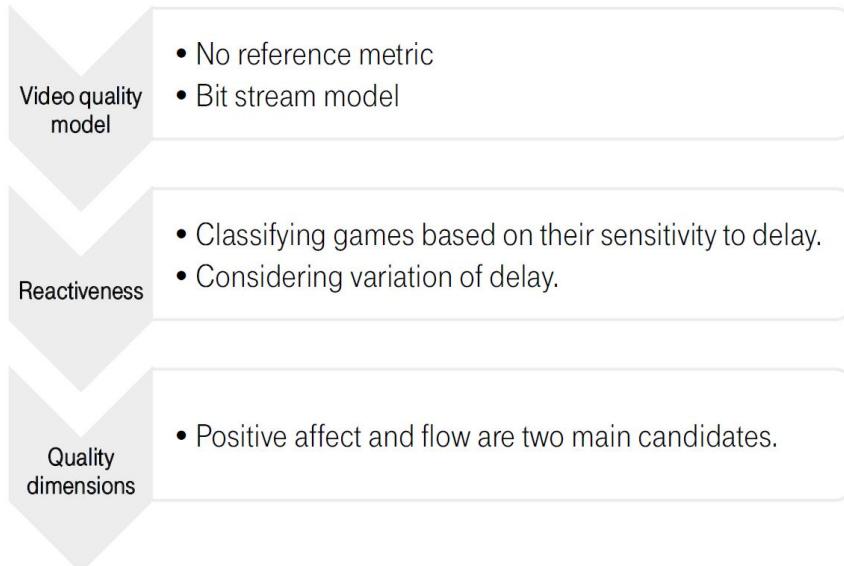


图 5.31 游戏 QoE 建模特点

本章参考资料：

- [1]<https://mp.weixin.qq.com/s/JFrZzdxGjUZNkdfDfQ0-Pw>
- [2]<http://www.52vr.com/article-2115-1.html>
- [3]<https://blog.csdn.net/tcpipstack/article/details/52024537>
- [4]<http://baijiahao.baidu.com/s?id=1594786358644717130&wfr=spider&for=pc>
- [5]<https://www.its.blrdrdoc.gov/vqeg/projects/immersive-media-group.aspx>

- [6]<https://www.its.blldrdoc.gov/vqeg/meetings/madrid-spain-march-19-23.aspx>
- [7]Brunnström K, Sjöström M, Imran M, et al. Quality of Experience for a Virtual Reality simulator[C]// Human Vision and Electronic Imaging. 2018.
- [8]Wang Z, Bovik A C, Sheikh H R, et al. Image quality assessment: from error visibility to structural similarity[J]. IEEE Transactions on Image Processing, 2004, 13(4):600–612.

第六章 相关国际标准组织

增强现实 AR 和虚拟现实 VR 的出现使得人们可以利用技术不断地去改进描述世界的方法。人们通过多种多样先进的设备，获得前所未有的“沉浸式体验”。所谓的“沉浸式体验”，是指用户身处于接近真实的虚拟世界，通过音视频技术，使用户完全投入情境当中。

近年来，VR 行业的发展态势良好，许多团队如 HypeVR、NextVR 始终致力于 VR 各环节的技术研究，Facebook、Samsung、HTC 等知名厂商也着眼于 VR，带来了许多先进、便捷的产品。一些体育赛事转播平台 BT Sport、Sky UK 等也已引进 VR 设备，为观众带来 360 度的观看体验。

根据高德纳咨询公司 2017 年度关于新兴科技的调查报告，VR 产业目前正处于复苏期，还需 2-5 年方可达到一个平衡发展的状态。目前的 VR360 视频还存在着分辨率低、头部运动范围小、观看设备庞大等局限性。VR 的最终目标是实现“6 自由度”的完全沉浸式体验，让人感到身临其境，并具有良好的交互感。

目前，多个国际组织正致力于沉浸式媒体的标准制定和研究，各组织致力于 VR 的不同方面，共同推进 VR 产业标准的发展，指导行业相关人员的技术研发。本章将主要介绍这些组织在 VR/AR 方面的近期标准工作进展。



图 6.1 VR360 标准概览

涉及沉浸式媒体标准制定的国际组织非常多，如 MPEG, 3GPP, Khronos, IEEE 等。还有一些工业论坛，如 VR-IF，代表相关产业界在产业方向上对沉浸式媒体发展的要求。

其中，MPEG 已经创造并且仍在制作媒体标准，适应工业需求，推动市场的发展。MPEG 将沉浸式媒体方面的工作放在一起统称为 MPEG-I (MPEG Immersive Media)。

6.1 MPEG-I

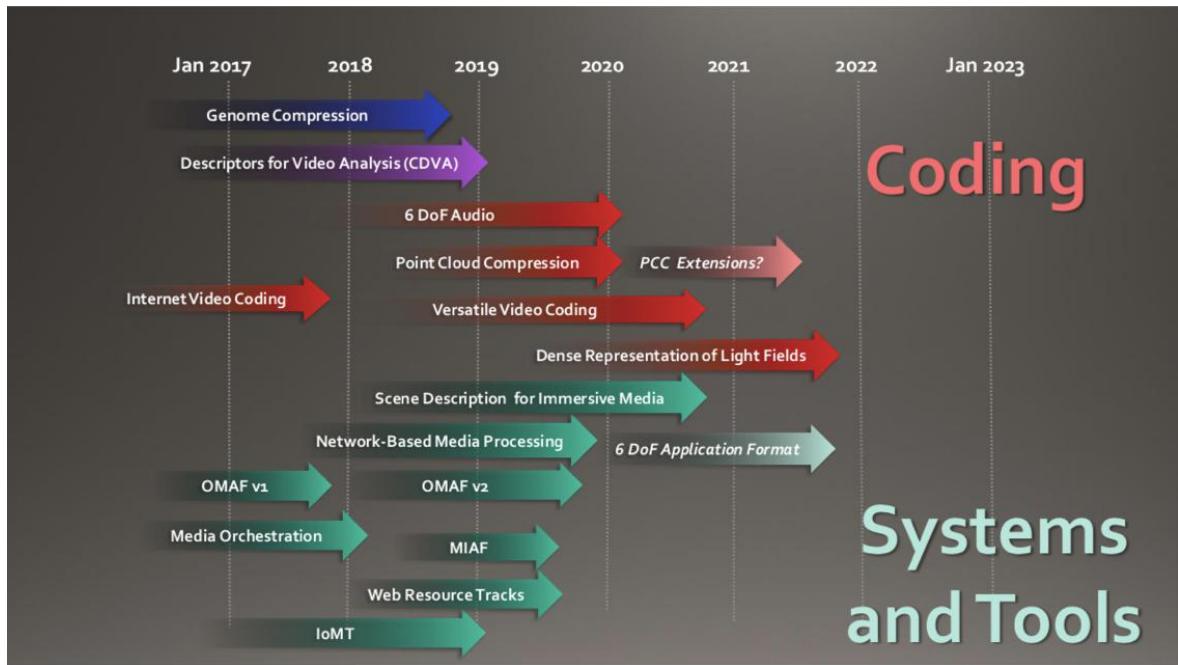


图 6.2 MPEG 对于沉浸式媒体的蓝图规划

MPEG 目前已经公布了其长期规划蓝图（约 5 年），如图 6.2，并收集了各大厂商的反馈和要求。其在沉浸式媒体网络视频编码、应用架构、服务调度等方面都已经有了相应的标准，未来将会把重点放在 6 自由度媒体、内容结合、VR360 直播与点播上。以下是 MPEG-I 标准中最主要的八部分内容：

- 沉浸式媒体架构；
- 全景媒体应用格式（OMAF）；
- 通用视频编码；
- 新型沉浸式音频编码；
- 点云编码；
- 沉浸式服务与应用元数据；
- 沉浸式服务与应用度量指标；
- 基于网络的媒体处理。

OMAF 框架下的 VR360 支持 HEVC、AVC 视频编码标准，MPEG-4 AAC、MPEG-H 音频编码标准，DASH、MMT 等多种流传输协议。在虚拟现实和增强现实的环境下，MPEG 期望给在复杂的，支持交互的场景中的媒体渲染定义规范，就像为 3 个 DoF 媒体定义的 OMAF 一样。

6DoF 应用格式将支持在 MPEG 容器中聚合和打包媒体，用于存储，下载，流媒体和广播分发。可以从场景图/场景描述文件中读取场景，或者可以从内容推断场景。

2018 年 7 月，MPEG-I 报告表示其已经完成了以下的部分工作，并仍需持续推进：

- 3DoF+ 和 6DoF 测试材料的征集
- 确定并核实 3DoF+ 和 Windowed 6DoF 的通用测试环境
- 扩展 3DoF+ 和 6DoF 的参考软件和配置
- Windowed 6DoF 和 全景 6DoF 的探索性实验
- 关于密集光场压缩技术的探索性实验

以下将简要介绍 MPEG 构建的沉浸式媒体标准架构，主要用于 6DoF 应用。

网络组件

控制和管理功能: 用于建立支持 MPEG 的客户端和支持 MPEG 的网络元素之间的通信, 或多个网络元素之间的通信。预计这种功能并不会在 MPEG 标准范围内, 除非有明确的要求。

MPEG 内容来源: 托管 MPEG 格式内容并且可以使用 MPEG 定义的协议访问的服务器 (集中式/分布式)。

MPEG 媒体网络感知/处理功能: 建立 MPEG 内容源或支持 MPEG 的客户端或两者的通信, 以支持沉浸式体验。

支持 MPEG 的客户端: 提供使用沉浸式 MPEG 内容的所有方法。

设备组件

- 底层 (5G, WiFi, IP)
- 媒体访问客户端 (DASH 客户端, MMT, 文件系统)
 - 提取基本流和元数据, 并使其可供解码器和应用程序使用
- 上行客户端
 - 上行媒体: 媒体编码器、内容交付协议
 - 元数据
- 解密
- 媒体解码器
 - 音频
 - 视频
- 应用程序/引擎
 - 场景图
 - 管理
 - 场景描述
 - 脚本
 - 定时跟踪
- 渲染
 - 音频
 - 视觉
- XR 功能
 - XR 输入
 - XR 显示
 - XR 合成
 - 传感器数据
- 捕获
 - 话筒
 - 相机
 - 其他

传统/第 1 阶段架构

第一阶段的架构基于 OMAF, 如图 6.3 所示。

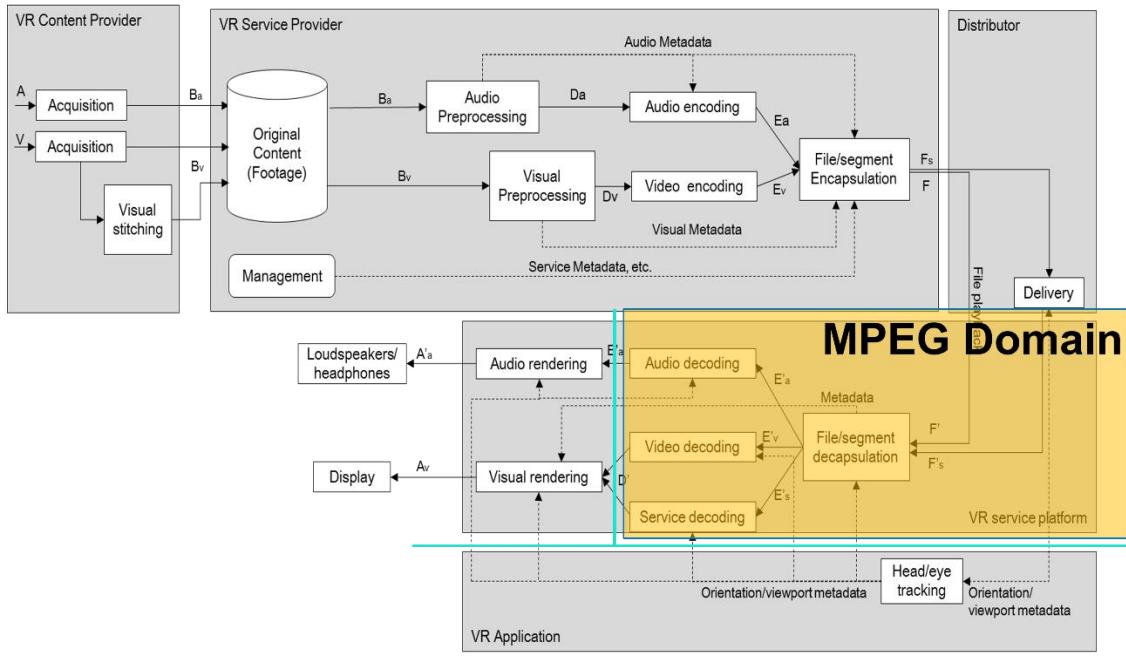


图 6.3 OMAF 架构和关键的 MPEG 域定义

从 MPEG 传统架构和系统模型开始，如 MPEG-2 TS, DASH, ISO BMFF 以及 MMT，系统标准主要提供：

- 1) 不同媒体流的定时和同步
- 2) 多路复用或基础媒体流的组合
- 3) 使用诸如随机访问内容的系统功能
- 4) 所需性能的描述
- 5) 流属性的描述
- 6) 能够在不同的网络场景下提供内容：适应，切换，错误恢复
- 7) 其他系统功能，如加密

用户/界面与传统内容“交互”的能力仅限于随机访问，搜索和组件选择。这种选择/交互不仅影响媒体播放，还影响媒体解码和传送。

在 MPEG-I 第 1 阶段中，通过提供多维视点，扩展了应用程序与媒体堆栈交互的体系结构。这种相互作用导致额外的动态和优化过程。除此之外，体系结构相对不变。

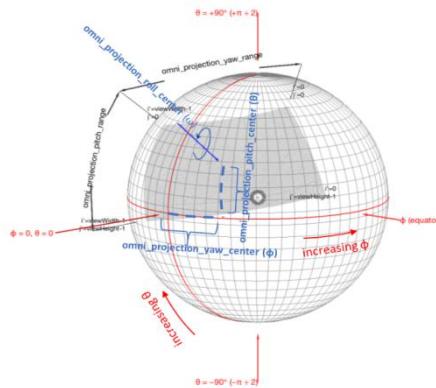


图 6.4 将视点相关的传感器数据添加到媒体消耗中

第 2 阶段架构

该架构预见到客户端通过不同的模式进行输入，以考虑 3DoF / 6DoF 甚至 AR / MR 等应

用。AR/MR 应用通常从本地摄像机和麦克风输入。

渲染由图形渲染引擎和 2D/3D 音频渲染引擎执行，其将解码的媒体资源合成在一起并呈现给用户。

该架构支持 MPEG 和非 MPEG 媒体资源。容器解析器提取有关资源和媒体时间线的信息以及任何嵌入或引用的媒体资源，并使它们在相关引擎中可用。架构支持多同形式的内容使用，例如简化的 2D 版本可以在简单的客户端中呈现，客户端也可以使用 3DoF, 3DoF+ 或 6DoF 版本。

为了描述展示的场景，可以使用场景图，也可利用替代格式提供场景图。例如，通过容器格式描述基本的呈现操作，可以简单客户端。其他场景描述文件也可以包含在容器中。

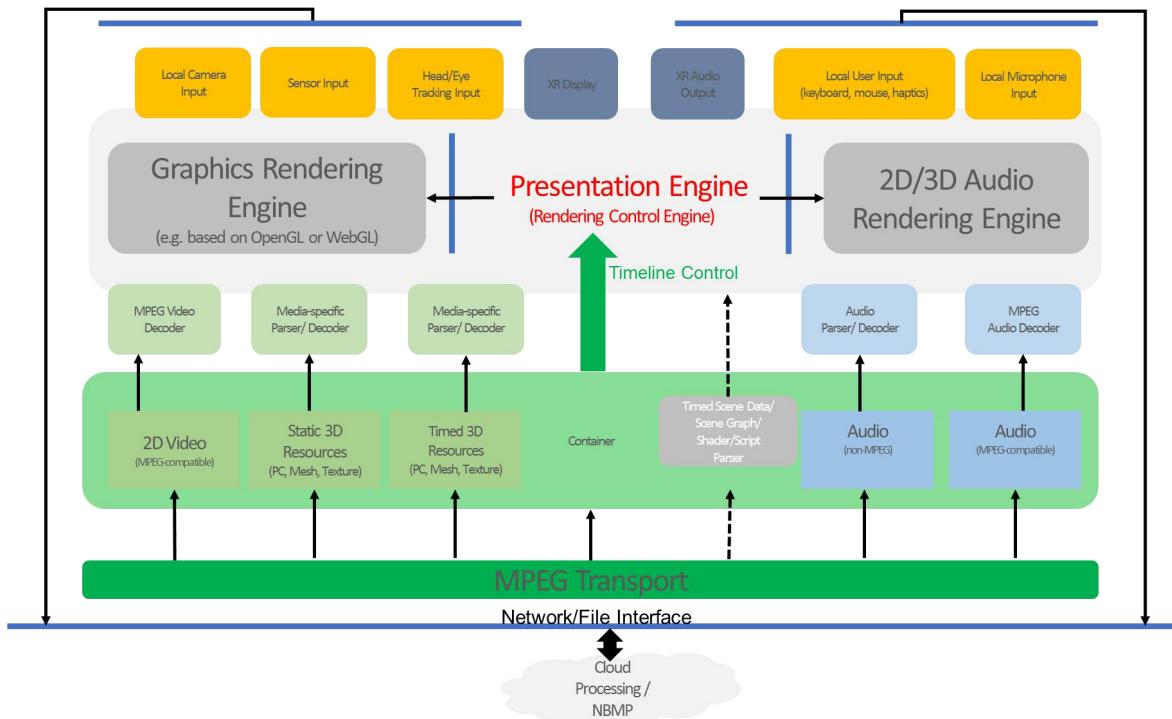


图 6.5 起草的第 2 阶段参考架构

以上为第 2 阶段的一般架构。目前 MPEG 已经考虑从网络、VR 社交性等方面形成一些针对性的替代方案。以下将展开讲述一种 MPEG 提出的以传输为首要目标的媒体转变思想。

MPEG-I 项目中的流媒体优先设计

1) 从广播和存储媒体到流媒体

目前，媒体行业技术创新主要由基于流的应用程序所驱动。对于传统的视频应用，例如视频流服务（Netflix, Hulu, Amazon Prime），情况确实如此，而对于沉浸式视频应用来说更是如此。由于沉浸式体验会占用大量带宽和存储空间，因此内容传递变得越来越个性化、多样化，以使数据流实时地适应消费者的确切需求。

在 MPEG-I 的背景下，新的媒体格式正在被标准化。对于这些格式，预期的应用/服务均是基于流的。此类示例包括 Google Light Fields, Samsung VR, Next VR 等。一对多分发系统（如广播）本身就不太适合这种沉浸式体验，因为单播技术例如 MPEG-DASH 这样的 HTTP 自适应流更适用于此类体验。虽然已经对广播 VR/360 内容进行了一些实验，但随着视频带宽和“个性”的增加，广播变得越来越不可能。当然，也可以想象一种混合模型，其中的一部分内容是一对多发布的，而其余部分仅传送至特定客户端。

这一转变有许多重要的含义：

1. 以发送方为中心的模型正在被以接收方为中心的模型所取代；
2. 很大一部分的传输数据对于接收端而言是唯一的；
3. 传输的数据需要实时适应接收端的动态需求，且必须以尽可能低的延迟提供；
4. 虽然总数据量增加，但最关键的点在于“基础可传输数据单元”的大小需要减小。

这种转变要求 MPEG 的设计原则也有所改变，这就需要考虑高效的流传输策略，或者压缩数据量。然而，目前 MPEG-I 的开发过程中缺乏对流媒体策略的考虑，同时对流媒体尽早的关注将大大增加采用这些未来 MPEG 技术的机会。因此，MPEG-I 提出改变设计原则。

2) 当前方法：从编码到传输的标准化

MPEG 最初是从编码格式的标准化开始的，之后定义了内容的存储，最后指定了传输格式。近年来，媒体已经由基于文件的应用程序和一对多传送驱动。上述传统方法在传输数据适配明确的带宽约束方面非常成功。

然而对于流应用，带宽约束不再是先验、已知的，并且不一定在小范围内，而是具有从例如 2Mbit/s 到 100Mbit/s 的连续频谱，甚至更多。此外，客户端设备很可能会受到传送网络质量的影响。而近年来出现的新型自适应技术一定程度上解决了这个问题，例如 MPEG-DASH，其中接收端负责在给定网络条件下最大化体验质量。

传统的方法是首先考虑压缩，然后再进行传输，这可能导致解决方案不是最优，因为：

- 在存储步骤中做出的决定可能会禁止某些流传输策略
- 流传输步骤可以简化为存储格式的线性传输
- 设计时可能不会考虑延迟等网络因素，然而这些因素对于任何实际部署都是至关重要的

3) 改进方法：流媒体优先的标准化

“流媒体优先的设计”是在考虑封装之前，着眼于媒体数据的流媒体策略。目的是确保文件格式设计能够实现沉浸式体验的高效流传输而不产生不必要的开销，这一原则允许在延迟和效率之间进行权衡。图 6.6 描述了这种标准化过程的三个主要步骤。

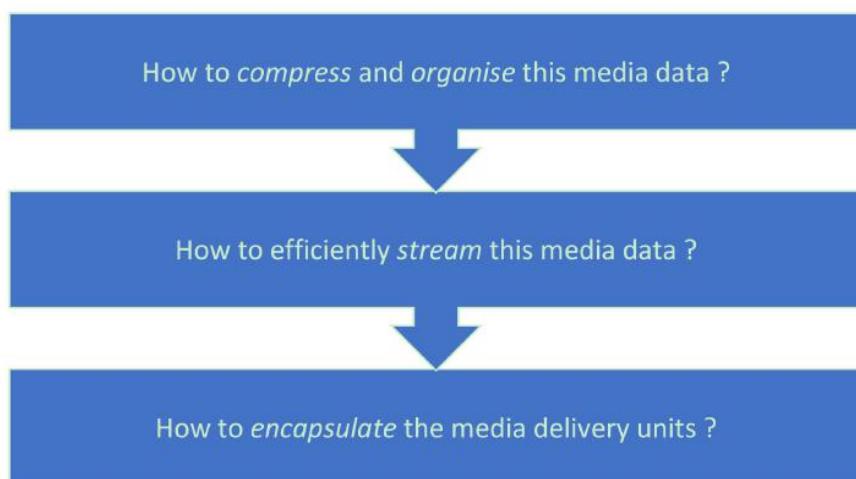


图 6.6 流媒体优化原则

流传输步骤将重点关注：

- 渲染媒体体验时的适应维度是什么？（例如，带宽，FoV，空间位置等）
- 相关的传输协议是什么以及如何充分利用它们？（例如，HTTP/1.1, HTTP/2, QUIC, WebRTC 等）
- 我们应遵循哪种策略以传输最少的数据并最大化 QoE？（例如，基于视角的流传输）

等)

- 如何应对网络延迟？（例如，预提取）
- 如何根据新的 MPEG-I 编解码器实现契合直播体验的端到端延迟？

目前，流媒体优先标准化的拟议范围包括：

- 短期
 - 提取 OMAF WD 的 Annex D 进行技术研究和整理
 - 为 Annex D 准备参考软件
 - 定义测试条件并进行实验以测量示例流条件下的传输比特率。
- 中长期
 - 高效利用 MPEG-I 媒体资源的现代网络协议的研究
 - 研究利用 MPEG-I 流媒体资源的当前 MPEG 技术（例如，ISOBMFF，DASH，MMT）的局限性，并可能在必要时启动新的流式格式

这种工作的最终目标是优化流式 MPEG-I 内容的 QoE，因而，可以期待一个独立于 OMAF 的规范、标准生成。

6.2 IEEE P2048

目前，IEEE 标准协会正在为虚拟和增强现实开发一系列标准。下表提供了关于 IEEE P2048 系列标准工作组的简要描述。在 2017 上半年，该工作组宣布了针对虚拟现实和增强现实的 8 个 IEEE 标准项目，目前已经拓展到 12 个相关项目。

IEEE P2048 部分	简要描述
P2048.1 设备分类及定义	此标准规定了关于 VR,AR 设备的分类和定义
P2048.2 沉浸式视频分类和质量度量 (P)	此标准规定了沉浸式视频的分类和质量度量，包括几个方面：360 度或 180 度，是否立体，观察点是否可移动，焦点是否可调整等等
P2048.3 沉浸式视频文件和流格式	此标准规定了沉浸式视频文件和流格式，以及格式所支持的功能和交互。
P2048.4 个人身份	此标准规定了在 VR 中验证（识别）人的身份的要求和方法。
P2048.5 环境安全	本标准规定了虚拟现实（VR）工作站和内容消费环境的建议。
P2048.6 沉浸式用户界面	此标准规定了在 VR 应用中启用沉浸式用户界面的要求和方法，以及沉浸式用户界面提供的功能和交互。
P2048.7 真实世界中的虚拟对象的映射	此标准规定了针对 AR,MR 应用的要求，系统，方法，测试和验证来创造和使用真实世界中的虚拟对象的映射
P2048.8 虚拟物体与现实世界的互操作性	此标准对于虚拟物体与现实世界的互操作性，在 AR,MR 应用中规定了要求，系统，方法，测试和验证
P2048.9 沉浸式音频的分类和质量度量	此标准规定了沉浸式音频的分类和质量度量。
P2048.10 沉浸式音频文件和流格式	此标准规定了沉浸式音频文件和流，以及格式支持的功能和交互。
P2048.11 车内的增强现实	此标准定义了协助车辆司机或乘客的增强现实（AR）的总体框架系统。
P2048.12 内容评级和描述	此标准定义的对于 VR, AR 和 MR 的内容评级和描述

图 6.7 IEEE P2048 12 个标准项目

表中项目将由 IEEE 虚拟现实和增强现实工作组负责，参与者包括设备制造商、内容提供商、服务提供商、技术开发人员和政府机构等等。目前，VR/AR 相关的技术正在以非常快的速度发展，而现有的开发标准仅涵盖了 VR/AR 领域的小部分，该工作组正全方面努力扩展，以达到满足技术领域的标准化需求。

6.3 IEEE P3333.3

IEEE P3333.3 标准主要的关注点在于头戴式显示器（HMD）的 3D 内容运动失真问题。该标准组织目前正通过对焦点失真的视觉反应、对镜片材料的视觉反应、对镜片折射率的视觉反应、对帧率的视觉反应等四个方面的研究提供一个用于解决“基于 HMD 的 3D 内容运动失真引起的虚拟现实（VR）失真”的技术指导。

P3333.3 BoD (Board of Directors)	
TG number	Part of Task
TG 1	Analysis of Human Factors (Medical Research included)
TG 2	S/W & Content (Best Practice Guide)
TG 3	Network (Wired & Wireless Latency, Handover)
TG 4	Display (OLED, LCD, Display Board)
TG 5	Sensors (Latency, Accuracy)
TG 6	Lens (Materials, Refraction Ratio)

图 6.8 IEEE P3333.3 WG 结构

6.4 Khronos



图 6.9

在 2017 年 GDC 大会期间，Khronos Group（科纳斯组织）公布 VR/AR 标准 OpenXR，为 VR 和 AR 应用程序定义了一个 API。在 OpenXR 中，应用程序和引擎使用标准化接口来询问和驱动设备，这样设备可以自我集成到一个标准化的驱动程序界面。同时，标准化的硬件/软件接口减少了碎片化，同时保留了实施细节，以鼓励行业创新。

由于不同的公司的系统和设备不同，这就导致了 VR/AR 设备和平台出现各种“分裂”的情况。OpenXR 的目的便是减少开发者们为支持不同设备的 API 而产生的麻烦。如果没有设立标准的 API 接口，各平台间的对接将会非常的繁琐和复杂。有了 OpenXR 标作为接口标准后，开发者只用写一次代码便能对接各个平台。Khronos 标准可用于增强用户界面，以及智能手机的游戏和应用中 3D 图形 API 的 OpenGL ES，用于异构并行计算的 OpenCL 以及用于 HTML5 的 3D 图形的 WebGL。

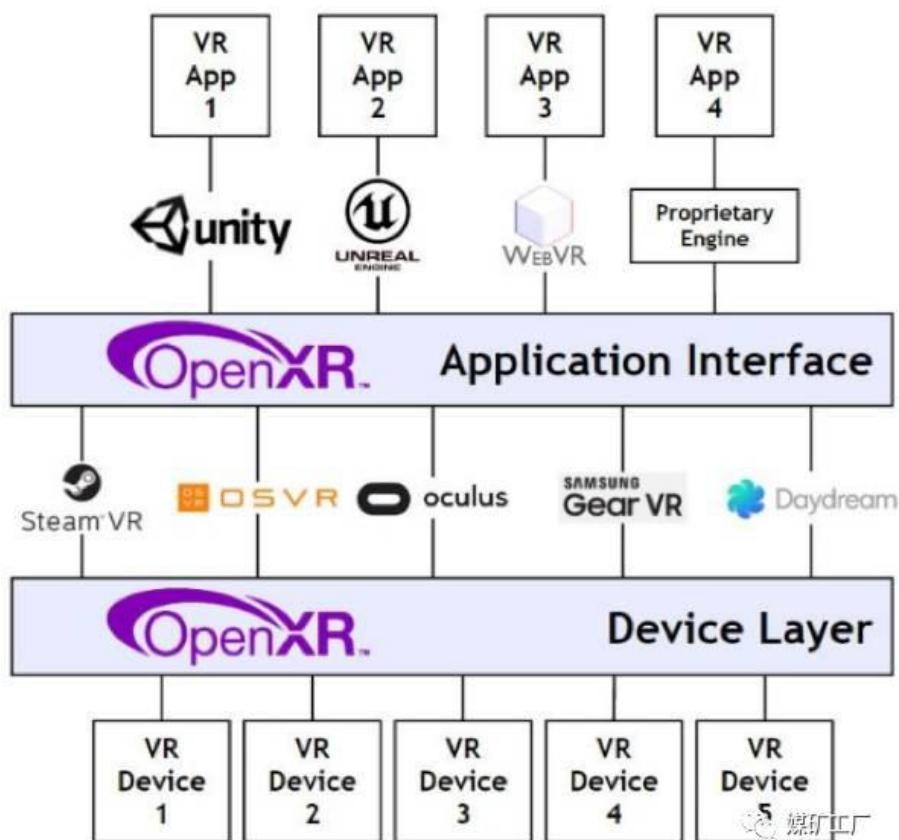


图 6.10 OpenXR 通用接口设计

6.5 WebVR

目前 VR 市场产品众多，无论在硬件还是内容服务上尚无法形成统一标准。而 WebVR 的出现，能够让诸多 VR 头显设备或是 VR 手机在获取内容的方式上统一。WebVR 提出了一个关于 VR 网络应用的开放性标准，即用户可以直接通过浏览器观看 VR 内容。在 2017 年 2 月，Google 已提出在 Chrome 浏览器上植入 WebVR，让 VR 体验更加便捷，图 6.11 为 Google 的 WebVR 实验室的设计截图。

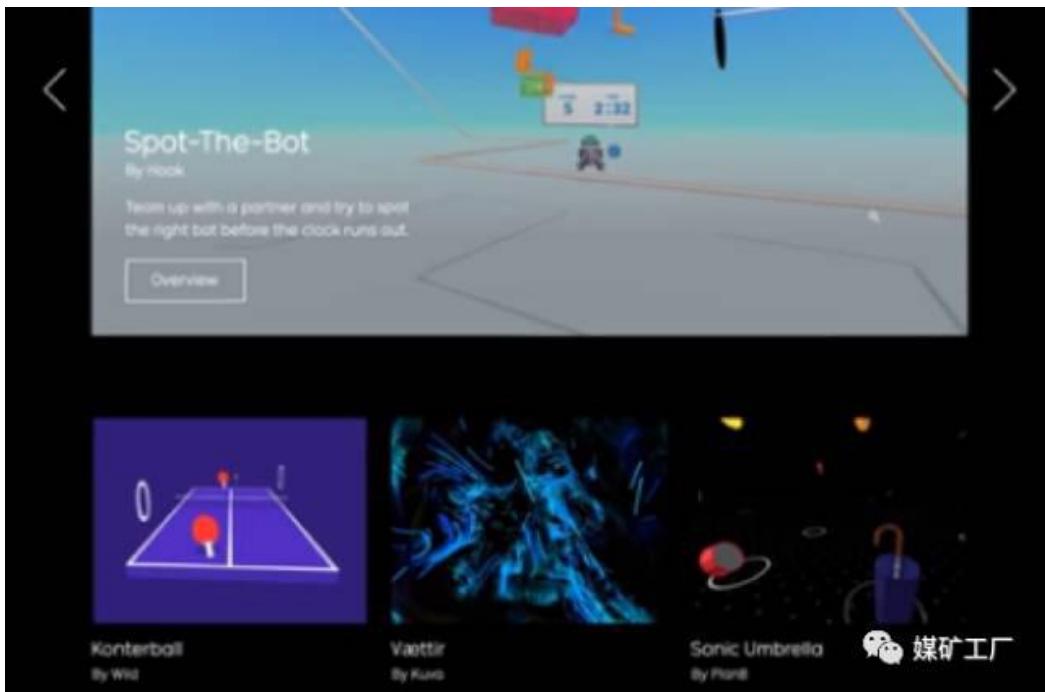


图 6.11 Google 的 WebVR 实验室的设计截图

Web XR 设备 API 规范提供 VR 和 AR 硬件的接口，使开发人员能够在 Web 上创建有趣舒适的 VR / AR 体验。它旨在完成后完全取代传统的 WebVR 规范。WebVR 的优势显而易见。它的开发门槛更低，一个普通 web 前端工程师就可以参与到 VR 应用开发中。它的跨平台性更强，可以跨设备终端、跨操作系统以及跨 APP 载体。其次，WebVR 开发快速、维护方便，可以随时进行调整，并且传播更加便捷。最重要的是，使用浏览器即可体验 VR，无需安装其他应用。2018 年 7 月，WebVR 提供了最新的 VR / AR 硬件接口。

6.6 3GPP



图 6.12 3GPP

3GPP SA4 关于虚拟现实 (FS_VR) 的研究项目结果被记录在技术报告 TR 26.918 中。该技术报告中主要包含以下七点：

- 现有的 VR 音频和视频内容制作工作流程和格式以及渲染；
- 一系列涵盖流媒体、会话、用户生成和遗留内容消费 VR 服务的用例；
- 双耳听音条件的音频质量评估；
- 视口独立和相关 VR 流媒体的视频质量评估；
- 延迟和 VR 音频/视频同步；
- 网络，内容和设备对 VR QoE 的影响；
- 差距分析和候选解决方案.

同时，3GPP SA4 启动了以下工作和研究项目：

- (1) 流媒体虚拟现实配置文件 (VRStream) 的工作项目，其目的是为 VR Streaming 用例集定义相关的媒体和协议启用者。
- (2) 沉浸式语音和音频服务 (IVAS_Codec) 的 EVS 编解码器扩展工作项目，总体目标

是开发用于身临其境的 4G 和 5G 服务和应用的单一通用音频编解码器。

(3) Live Uplink Streaming (FLUS) 框架工作项目，其目的是为点对点定义上行链路直播流媒体（例如 360 视频，VR，UHD，多声道音频）的框架。

(4) 针对 VR 音频的 3GPP 编解码器的研究项目 (FS_CODVRA)，其目标是评估现有 3GPP 音频编码能力是否适合启用 VR 服务，并提供如何使用和配置现有编解码器以提供最佳 VR QoE 的推荐。

(5) 关于 VR 的 QoE 指标的研究项目 (FS_QoE_VR)，其目标是调查可能需要由客户端向网络报告以评估 VR 用户体验的 QoE 参数和指标。

VR Streaming

3GPP 最终确定了 TS 26.118 (VR Streaming 的媒体配置文件) 的技术工作。最新版本 TS26.118 1.1.0 可用。

- 该规范预计将于 2018 年 9 月获得批准
- 定义视频和音频的操作点（基本流和渲染要求）以及媒体配置文件（包括文件格式和 DASH 约束）
- 视频
 - 操作点
 - 基本 H.264 / AVC: 带有 ERP 的 H.264 / AVC HP@L5.1
 - 主要 H.265 / HEVC: H.265 / HEVC MP10@L5.1, ERP, RWP, 立体视觉; 从 OMAF 中选择工具
 - 灵活 H.265 / HEVC: H.265 / HEVC MP10@L5.1, 增加了立方体投影和 HDR; 高达 120 fps
 - 媒体资料
 - 基础视频: 基于 H.264 / AVC OP, 单流 HEVC, 无视点优化
 - 主视频: 基于 H.265 / HEVC OP, 样本入口 hvc1, 单个或多个独立的自适应集, 单个流呈现
 - 高级视频: 基于灵活 H.265 / HEVC OP, 样本入口 hvc1/hvc1, 提供单个或多个相关的自适应集, 单个和多个流呈现, 允许 tiling 等。
 - 所有配置文件都可以作为 OMAF 视频配置文件的子集被提供。
- 音频
 - 测试了 4 种候选方案; 具体结果记录在技术报告中
 - 同意纳入兼容 OMAF 的 MPEG-H 音频
- 元数据: 支持在 2D 屏幕上渲染 360 体验。
- 在系统级别, 完成 PSS 和 MBMS 服务的集成。这意味着广播 VR 是可能的。

6.7 DVB

DVB(数字视频广播)是电视、广播和技术公司的联盟，旨在为数字电视和其他广播技术创建一套开放的技术标准。在虚拟现实(VR)发布报告后，DVB 为 VR 内容设定了新的标准。在 2017 年年中，DVB VR 活动从 VR (CM-VR-SMG) 的商业模块 (CM) 研究任务被推广到了 CM-VR 官方团队。CM-VR 的总体目标是提供商业需求，传递给相关的 DVB 技术模块 (TM) 小组，根据 DVB CM 的规定开发针对 DVB 网络上的 VR 内容交付的技术规范。



图 6.13 DVB

CM-VR 首先将重点放在系统上，针对 DVB 集成式接收器/解码器（IRD）以及下一代 IP 连接设备，利用 DVB 宽带和广播网络向广播公司提供 VR 内容。考虑到现有的技术和现实的部署方案，CM-VR 将打算提供“全景/ 3DOF +”的视听体验。

目前，DVB 定义了 VR 在商业上获得成功的三大要素：技术、运动晕眩以及内容，其中技术方面包括制作、传输以及 VR 显示。

由于 VR 技术的规模和复杂性，DVB 在第一阶段将研究 VR 和 360 度 VR 设备的体验，第二阶段将考察更为复杂的 VR 设备领域。CM-VR 的目标是能够获得 CES2018 规定的可用的稳定要求，然后将这些标准提交 DVB 商业模块和指导委员会批准。无论如何，DVB 显然都会考虑到其他 VR 标准化组织和行业组织所做的工作，特别是 MPEG，VR 行业论坛和 3GPP。CM-VR 未来将协调不同组织的工作而做出自己的贡献。

一项调查问卷于 2018 年 2 月 26 日启动，重点是 VR / 360 / 3DOF 内容，目标是：

- 确保存在商业需求
- 关注使用情况，根据用户兴趣确定优先级

到目前为止，已收到并处理了 7 份答案：2 家技术提供商，1 家广播公司，1 家广播网运营商，1 家专业设备制造商，2 家消费电子制造商

然而在 2018 年 7 月，考虑到对 CM-VR 组的支持程度，SB 决定暂停 VR 工作六个月，并在 2019 年 2 月在 SB91 上重新考虑该主题。

6.8 VRIF



图 6.14 VRIF

VRIF 建立的目的是为 VR 提供一个广阔的市场，维护消费者、内容设备制造商、服务提供者、广告公司等多方利益，进一步广泛提供高质量的音像虚拟现实体验。目前的重点是为分配音频和视频内容服务实现高质量，可互操作的体验。VRIF 最新发行的 VR 指南已在 CES 2018 期间公布，讲述了 VR 内容创作、传输、安全性、交互性等技术细节，进行了少量更正和其他说明。

- 即将推出的版本
 - 关于直播的工作已经开始
 - 开始着眼于 VR 和 HDR 的结合
 - 将 Presentation API 添加到指南（渲染）

- 开始解决文本和字体问题
 - 安全性：VR 内容中的水印
 - 测试和交互操作：正在逐步推进
- VRIF 指南的初始版本侧重于具有三个自由度的 360° 视频传输系统（3DOF），并包含：
- 基于 ISO MPEG 全景媒体格式（OMAF）的跨行业互操作点文档
 - VR360 内容的最佳实现方案，重点是人体因素，如晕动症
 - VR360 流媒体的安全注意事项，侧重于内容保护，同时也关注用户隐私。

6.9 QoE: QUALINET, VQEG, ITU-T

ITU 在 QoE 方面的一些工作在之前章节中已经有所介绍。目前，ITU-R WP 6C 工作组通过工作草案描述了一种原型 HMD，其空间分辨率为 $8K \times 4K$ ，完整 360° 图像的空间分辨率为 $30K \times 15K$ 。此外，他们已经开始研究高级沉浸式视听（AIAV）系统的参数值，例如图像分辨率，投影映射方法以及线性 360° AIAV 程序等。ITU-T SG12 Q13/12 目前给出了与沉浸式媒体相关的以下工作项目：G. QoE-5G（5G 网络新型服务的 QoE），G. QoE-AR（增强现实（AR）的 QoE），G. QoE-VR（VR 服务的 QoE）和 P. 360-VR（或 G. 360-VR，用于 HMD 上 360 视频的主观测试方法）。接下来将对后两个项目进行简要介绍。

G. QoE-VR

在该项目中，华为贡献了一份“G. QoE-VR 基准的更新版本”。主要增加的是关于 VR QoE 内容指标的新部分，并在 Krakow Q13/12 会议上与 VQEG 一起讨论过。

讨论总结如下：

- 来自 TU-Ilmenau 的代表认为这份文件包含的技术水平过高，建议修改它的结构，这一点得到了一致的认可。
- 之后再次提交了新的更新版本并得到同意，成为新的基准。

该文件中介绍了当前主要的 VR 应用场景：直播、点播、游戏、社交与购物。而影响 VR 的因素目前被归纳为：

- 人为影响因素
 - 视听
 - 模拟器疾病
- 系统影响因素
 - 内容相关因素：空间音频、空间深度（3D）、时空复杂性、类型等
 - 媒体/编码相关因素：音视频编码器、比特率、分辨率、帧率、音频采样率、编码延迟等
 - 网络/传输相关因素：丢包、延迟、带宽、波动等
 - HMD 相关因素：解码器性能、头部跟踪延迟、自由度、FoV、显示分辨率、刷新率等
- 背景影响因素
 - 物理背景
 - 时间背景
 - 社会背景
 - 任务前后关系

G. 360-VR

- 华为同样提交了一份“G. 360-VR 基准文件”，并收到了 TU-I1menau 的一些建议。
- TU-I1menau 提出了“使用 HMD 进行 360 度视频 QoE 评估的主观测试方法”。
 - 它比较了两种主观评价方法：修正后的 ACR 和 DSIS 用于 360 度视频 QoE 评估。
 - 结果显示 M-ACR 在统计学上更可靠，并且用户不太容易发生模拟器疾病。
 - 因此建议将 M-ACR 用于评估 360 度视频，特别是短序列，如 10s。
 - 该建议已被同意纳入基准。

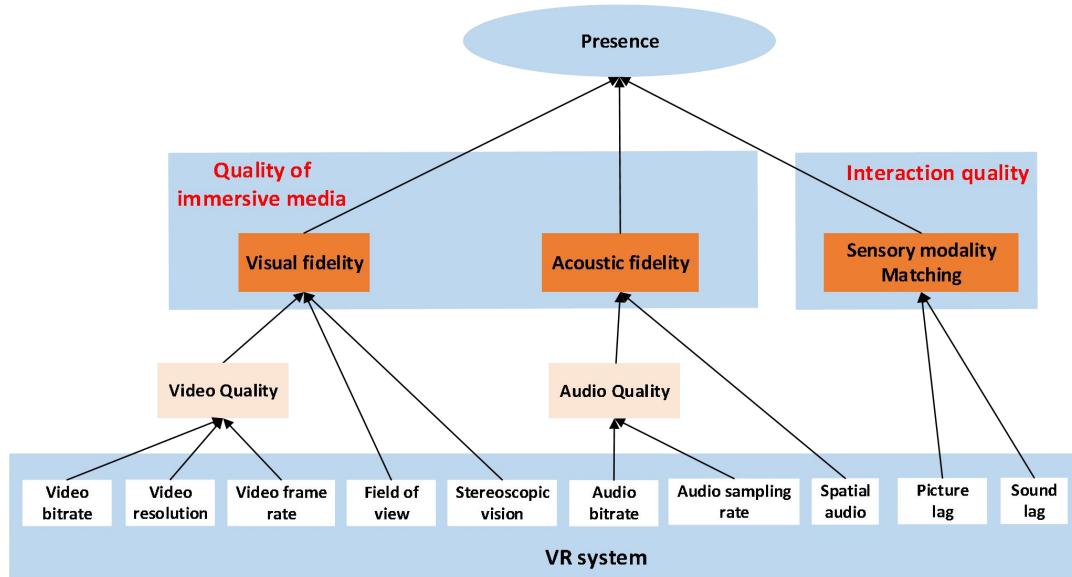


图 6.15 G. 360-VR 的框架

G. 360-VR 框架

- 源端条件
 - 视频性能
 - 应允许单视场和立体内容。
 - SRC 的质量也应尽可能相同。
 - 建议使用 4K 或更高分辨率的视频序列，以避免在 VR 显示器上放大的原始 VR 内容的低分辨率导致体验极度不佳的情况。
 - 音频性能
 - 可以使用立体声和空间音频。
 - 需要更多调查。
 - 互动性
 - 360 VR 应用是弱交互式 VR，用户在虚拟环境中被动地体验预先拍摄的内容。
 - 360 VR 的交互体验主要体现在 MTP 延迟上。
 - 持续时间
 - 使用持续时间为 10 秒至 5 分钟的序列。
 - 由于 10 秒至 20 秒非常短，因此建议使用如 M-ACR 的方法。

- 测试方法
 - 选项 1 – ACR
 - 选项 2 – M-ACR
 - 选项 3 – Double Stimulus Impairment Scale 测试法
 - 其他方法

M-ACR 和 DSIS 测试法流程如下所示：



图 6.16 M-ACR 法流程



图 6.16 DSIS 法流程

- 测试环境

测试时大致分为两种环境：

- 受控环境
- 公共环境

受控环境应代表一种不会分散人注意力的环境，在这种环境中，测试者可以合理地使用测试设备。该条件下，测试应在噪声隔离环境中进行，因为噪声是影响测试的主要环境因素。

公共环境应代表一种会让人分心的环境，该环境具有可能影响用户的噪声或其他因素。

2018 年 7 月，QUALINET 与 QoMEX 共同开展了年度会议，更新了沉浸式媒体的相关部署，特别是关于“沉浸式媒体 Qualinet-VQEG 联合团队 (JQVIM)”和“沉浸式媒体体验 (IMEx)”的任务。

此外，VQEG 启动了沉浸式媒体工作组 (IMG, Immersive Media Group)。IMG 整理了来自各个研究团队的可用全景图像/视频集，如表 6.1、6.2 所示。

表 6.1 360 图像集

数据集/ 提供者	图片 数量	分辨率	格式	附加材料	(资料) 注释	URL
Salient360 Université de Nantes / Technicolor	98	从 5376 × 2688 至 18332 × 9166	经纬图 投影	眼部/头部 跟踪数据	Check: Y. Rai, J. Gutiérrez, and P. Le Callet, “A dataset of head and eye movements for 360 degree images,” in Proceedings of the 8th ACM Multimedia Systems Conference	salient360 @univ-nant es.fr

					(MMSys), Jun. 2017.	
Stanford University, Universidad de Zaragoza, University of California Berkeley	22	8192 × 4096	立方体投影	眼部/头部跟踪数据	V. Sitzmann <i>et al.</i> , “Saliency in VR: How do people explore virtual environments?,” <i>IEEE Trans. Vis. Comput. Graph</i> , 2018.	https://vsitzmann.github.io/vr-saliency/
Berkeley V-Sense, Trinity College Dublin	22	4096x2048	经纬图投影		A. De Abreu, C. Ozcinar, and A. Smolic, “Look around you: Saliency maps for omnidirectional images in VR applications,” in <i>2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)</i> , 2017, pp. 1 – 6.	https://v-sense.scss.tcd.ie/?tag=saliency-maps
Nanjing University, Vision Lab, Immersive images	15	4096×2160			MOS	http://vision.nju.edu.cn/index.php/database/item/64-im-images
SUN360 MIT	67,583 (来自英特网)	多种分辨率, 最高为: 9104x4552	经纬图投影		J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, “Recognizing scene viewpoint using panoramic place representation,” <i>Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.</i> , pp. 2695 – 2702, 2012.	http://people.csail.mit.edu/jiao/SUN360/main.html

Laboratory for Image & Video Engineering	10	4096×2160	经纬图 投影		“To understand the joint impact of compression and spatial resolution”	http://live.ece.utexas.edu/research/quality/immersive_images/
Omnidirectional HDR consumer camera dataset, MMSPG EPFL	43	5476x2688	经纬图 投影	速率、跟踪和切换信息	A.-F. Perrin, C. Bist, R. Cozot, and T. Ebrahimi, “Measuring quality of omnidirectional high dynamic range content,” <i>In Applications of Digital Image Processing XL</i> , 2017, p. 38.	https://mmspgepfl.ch/360hdr-consumercamera

表 6.2 360 视频集

提供者	视频数量	分辨率/帧率	时长	格式	附加材料	(资料)注释	URL
IMT Atlanti-que (from Youtube)	7	3840×2048 / 25, 29. 97, 30, 60 fps	70 s	经纬图 投影	头部数据	X. Corbillon <i>et al.</i> “360-Degree Video Head Movement Dataset,” <i>ACM MMSys’17</i> .	http://dash.ipv6.enstb.fr/headMovements/
National Tsing Hua University (from Youtube)	10	4K	1 min	经纬图 投影	头部数据	W-C. Lo <i>et al.</i> , “360 Video Viewing Dataset in Head-Mounted Virtual Reality” <i>ACM MMSys’17</i>	https://dl.acm.org/citation.cfm?id=3192927
Tsinghua University	18	多种分辨率：4K, 2560x1440, ...	2-8 min	经纬图 投影	头部数据	C. Wu <i>et al.</i> , “A Dataset for Exploring User Behaviors in VR Spherical Video Streaming”, <i>ACM</i>	https://wuchlei-thu.github.io/

						MMSys' 17.	
Stanford University – Virtual Human Interaction Lab (from Youtube)	73		29–668s		头部数据 唤醒速率	B. Li, et al.. “A Public Database of Immersive VR Videos with Corresponding Ratings of Arousal, Valence, and Correlations between Head Movements and Self Report Measures”, <i>Frontiers in Psychology</i> , Dec. 2017	http://vhill.stanford.edu/360-video-database/
Universidad de Zaragoza , Stanford University	216	4K	6+6 s (2 shots)	经纬图 投影	眼部/头部跟踪数据	A. Serrano et al., “Movie editing and cognitive event segmentation in virtual reality video”, <i>ACM TOG</i> 2017.	http://webdiis.unizar.es/~aserrano/projects/VR-cinematography
Nanjing University, Vision Lab,	2	8K					http://vision.nju.edu.cn/index.php/data-base/item/83-pa-videos
Inter-Digital	6	8K	60s (4x), 300s (2x)	经纬图 投影		MPEG internal document but content license allows usage for research and publications but not for commercial use	

Wuhan University	48	从 2880×1440 至 7680×3840	从 20s 至 60s		头部数据	M. Xu, C. Li, Z. Wang, and Z. Chen, “Visual Quality Assessment of Panoramic Video,” pp. 1 – 12, Sep. 2017.	https://github.com/Archer-Tatsu/head-tracking
Technische Universität Ilmenau	20	4K	20s	经纬图投影	头部数据 + SSQ 结果	S. Fremerey et al., “AVTrack360: An open Dataset and Software recording people’s Head Rotations watching 360° Videos on an HMD.”, ACM MMSys 2018.	https://github.com/Telecommunication-Telemedia-Assessment/AVTrack360SRCs : 请联系: stephan.fremerey@tu-ilmenau.de

而在 2018 年召开的 VQEG 会议上, IMG 同意:

- (重新) 开始评估沉浸式媒体质量的联合工作。
- 不仅关注“娱乐”360 视频的视听质量, 还将范围扩大到:
 - “带有任务的视频”
 - “具有互动性的视频(或 VR)”
 - 不仅仅评估“视频质量”

据此, IMG 对其工作范围和具体应用场景进行了初步的归纳和总结。应用场景为:

- 单向应用场景(360 视频)
 - 娱乐
 - 培训(面向任务)
- 双向应用场景(交互式)
 - 机械控制(人与机器)
 - 私人交流(人与人)

结合实际约束条件, 细化的应用实例如下表所示:

表 6.3 IMG 应用实例总结

应用实例		自由运动	语义导航	带有任务的评估	交互性
单向	娱乐	有	+/-	无	无
	培训	有	有	有	无
双向	机械控制	有	有	有	有限制的
	私人交流	有	有	有	自由交谈

而其工作内容包括:

- 创建源内容的参考数据库
- 表征/分类源内容
- 定义要衡量的质量因素以及具体做法（主观方法）
- 建立测试条件/视频损坏对于 QoE 影响的评估方法
- 生成上述 4 点主观结果的数据集。

对于质量评估，IMG 初步考虑从视听质量、呈现效果、晕眩程度、效率等方面进行考量。测试条件包括视频捕获、压缩与传输、终端显示、延迟等。QoE 则包括沉浸感，参与感，存在感等。

第 1 阶段：全向视频的视听质量

IMG 第 1 阶段的目标是在 VR360 的视听质量方面，以传统视频指标如 PSNR、VQM、VMAF 为参考，达到与传统视频相同的成熟度。具体目标有以下几点：

- 定义评估 VR360 视听质量的主观方法，以某种方式等同于 ITU-T P. 910, ITU-R BT. 500 等对传统视频质量的评价方法。
- 收集一组“要求严格但不过分严格”的参考序列。
- 基于一系列合理的视频损坏条件，共同创建一组主观评估序列，用作可重复研究的参考，以及客观指标的训练集。
- 提出客观指标来模拟 360VR 视频的视听质量。

IMG 要求参考序列持续时间必须为 60 秒（可能不是，但应为一段较长的时间）。原始序列必须具备高质量：

- 采用中高端相机系统录制。
- 视频可以是：
 - 单视场序列。
 - 将左眼和右眼视图独立编码为文件的两个完整流。
 - 将左眼和右眼视图编码为单个数据帧的 Top-bottom 流。
 - 无缝拼接的全景视频。
 - 高分辨率的经纬图投影视频：8K, 8K UHD, 6K, 4K, UHD, 3K, HD。
 - 高帧率。用于测试的高质量视频应支持不同的帧速率。预期样本应高于每秒 30 帧：30 fps, 60 fps, 90 fps, 200 fps。
 - 定向音频。可能为：四，五，七，八扬声器配置，高保真度立体声响配置。
 - 未压缩或具有感知无损压缩：MPEG-2 TS, MP4, 原始视频, OpenEXR 文件。
 - 具有与视频捕获相关的其他元数据信息：深度映射视频轨道，其他传感器信息。

而在序列组中，视频损坏条件应涵盖视频数据流的预期情况，同时这种损坏必须是各向同性的。这样，视频缺陷和用户行为之间的交互就将减少。

要考虑的损坏包括：

- 几何变化：投影，分辨率等。
- 切换到单视场视频。
- 视频压缩。
- 自适应流式传输的影响（如卡顿）。

最终视频并在 3 DoF 消费设备（手机）上播放，因为这是 VR360 视频较为常规也是预期的应用场景。

此阶段工作的最终目的是利用该评估方法检测来自不同实例的代表性内容：广告，电影，体育，新闻，纪录片，教育等。

第 2 阶段：沉浸式媒体的体验质量

第二阶段应侧重于建模和度量沉浸式媒体，以更好的方式来表征参与度，沉浸感，不适感和其他影响。

具体而言就是研发一种标准化的衡量方法，需要考虑以下方面：

- 基于任务的评估，可用于评估交互性，如 ITU-T P. 805 或 P. 920。
- 基于行为的评估，替代基于问卷的评估。
- 提供对测量内容的精确定义，以便能够为其研发方法。

为开展相关工作，IMG 已为沉浸式媒体质量评估提供了明确的框架和术语定义。其中，沉浸式媒体框架涉及到：

- 硬件组件（显示器，控制器等）
- 信号
- 交互水平
- 运动自由度
- 渲染自由度

部分术语定义如下表所示：

表 6.4 沉浸式媒体术语定义

Binocular Disparity	左右眼看到的图像位置的差异，或者显示器显示的左右视图上的差异。
Earcon	用于表示特定事件或传达其他信息的简短、独特的声音。（来源：维基百科）
Head-Motion Parallax	从不同位置或方向观察物体位置的位移或差异。
Motion-to-High-Quality Latency	头部运动和在头戴式设备中以高质量显示内容所花费的时间。
Overlay	基于 360 度视频内容的视觉媒体渲染。
Transparency	视觉媒体的一个或多个叠加，其中叠加区域在视觉上是同质的，并且叠加和 360 视频内容的可见度可以是任意选择的，并不一定是“二选一”。
Viewing Space	用于观看的 3D 空间，其中可以进行图像/视频渲染和 VR 体验。
Viewpoint	用户观看场景的点；它通常对应于摄像机位置。轻微的头部运动并不意味着不同的观点。
Visual Media	视频，图像和文字。

目前，IMG 已与多个实验室展开合作，相关实验室/联系人及其测试设备如下所示：

- 诺基亚贝尔实验室，Pablo Pérez
 - HMD：三星 GearVR（带 Galaxy S8 / S9），Oculus Rift，HTC Vive
 - 计算机：具有高处理能力的台式机/笔记本电脑（nVIDIA GPU 等）
 - 生理/行为测量设备：ECG

- COMLAB (Roma TRE 大学) , Federica Battisti
 - HMD: HTC Vive, Microsoft Hololens
 - 计算机: 具有高处理能力的台式机/笔记本电脑 (nVIDIA GPU 等)
 - 生理/行为测量设备: 配备传感器的椅子 (正在开发)
- 分布式和互动系统 (DIS) , CWI, Francesca De Simone
 - HMD: 2 Oculus Rift, 1 OSVR
 - 计算机: 每个 HMD 一台服务器
 - 采集设备: 8 个相机, 2 个 kinects
- IPI-LS2N (Université de Nantes) , Jesús Gutiérrez
 - HMD: 2 个 HTC Vive, 1 个 Hololens, 1 个 Metavision
 - 计算机: 每个 HMD 一台服务器 (nVIDIA GPU 等)
 - 生理/行为测量设备: 眼动仪 (适用于 HTC Vive, Hololens 等), 脑电图, 心电图, GSR,
 - 采集设备: Ricoh Theta, Lytro Illum
- Vaader-IETR (INSA Rennes) , Fang-Yi Chao
 - HMD: HTC Vive
 - 计算机: 用于 HMD 的服务器和具有 nVIDIA GPU 的台式机等。
 - 生理/行为测量设备: HTC Vive 眼动仪

6.10 DASH-IF

DASH-IF 目前正在研究低延迟 DASH, 这与 Live VR 服务相关, 并且可能包括具有基于 DASH (VR) 分发的常规广播同步的内容。

6.11 CTA

2018 年 7 月, CTA 发布了 CTA-2069, 描述了增强和虚拟现实技术的定义和特征, 介绍了新兴消费级技术的各种术语。

在更早期, CTA 在 AR/VR 方面建立了第一个标准工作组。涉及内容包括:

- 虚拟现实 - 消费者体验和期望, 该研究调查了美国消费者的 VR 体验, 以深入了解消费者所接触的内容及其内容偏好。
- 消费者情感: VR 店内演示: VR HMD 及内容, 本研究将为视频内容的开发和分发策略以及 VR 产品的未来提供行业指导。
- 增强现实与虚拟现实: 消费者情感, 消费者反馈报告可以确定消费者对 AR 和 VR 技术及其各种用例的认知和感知。

6.12 SMPTE

SMPTE VR/AR 研究组于 2018 年 2 月 28 日成立, 旨在研究图像/声音捕获和发布的标准化方法的当前和未来需求, 以创建 VR/AR 的分发和显示系统。另一个目标是研究可能的标准应用场景。

具体来讲，VR 和 AR 内容有许多不同的捕获方法，文件格式，显示系统和后期制作方法。该小组要解决的问题是确定是否需要将这些方法标准化，以实现更简单、容易的替换。小组目前正在研究用于生产和后期制作的 VR 和 AR 系统，并创建一份报告，记录当前系统，相关现有标准和新标准的建议，并对现有标准和所需标准进行差距分析。

6.13 ETSI

ETSI 于早期启动了增强现实新组织，特别是增强现实行业规范组（ARF ISG），“在工作初期阶段，ARF ISG 希望听取一些 AR 行业用例，部署（试点）AR 服务时遇到的障碍以及互操作性要求。”

6.14 SVA

流媒体视频联盟（SVA）于 2016 年底成立了 VR/360 度视频研究小组。他们目前的工作是记录 360 视频市场的相关技术和经验。除了评估传统视频服务的 CDN 性能外，SVA 还希望包含 VR360 内容，以便了解延迟因素和 CDN 对 VR360 传输的影响。

小结

“沉浸式”媒体时代与 2D 时代相比，其带来的信息量是巨大的。采集呈现，存储与传输，对于技术实现来说都是很大的挑战。国际标准组织在压缩编码方面还是会发挥很大的作用。OMAF 等标准已经基本实现 3 自由度的视频，而 6 自由度的视频还需要更多发展的空间。对于 Immersive Media 来说，提升体验感，推动更多人用是当下最关键的。这一系列工作都需要标准工作来督促与推进。当下，越来越多的组织和企业都加入到制定沉浸式媒体活动的标准中，共同推进这项工作的创新与进步。

本章参考资料：

- [1]Christian Timmerer. Overview of standards activities related to immersive media (v2). ISO/IEC JTC1/SC29/WG11 MPEG2016/N17136. October 2017, Macau, CN.
- [2]<https://mp.weixin.qq.com/s/IPbXbgoFq8U5KgIk9zVNAA>
- [3]State of the Art, State of the Industry and State of the Standards, ACM Multimedia System Conference - Amsterdam, 12 June 2018
- [4]Virtual Reality (VR) media services over 3GPP. (Release 15). 3GPP TR 26.918 V15.0.0. 2017, 09.
- [5]Timmerer C. Immersive Media Delivery: Overview of Ongoing Standardization Activities[J]. 2017, 1(4):71-74.
- [6]<https://lists.aau.at/mailman/listinfo/mpeg-i-visual>
- [7]https://www.itu.int/ifa/t/2017/sg12/exchange/wp3/q13/G.360-VR/WD11_Huawei-7-Proposed baseline for G.360-VR-v2.docx
- [8]https://www.itu.int/ifa/t/2017/sg12/exchange/wp3/q13/201802_Interim_meeting_Geneva/WD14- Restructured version of G.QoE-VR baseline.docx
- [9]<https://www.its.blrdoc.gov/vqeg/projects/immersive-media-group.aspx>
- [10]<https://docs.google.com/document/d/1xLxVeXYCeGRHfPMWyiIo0ELvR00pUsHGdLE61WYhJOM/edit?usp=sharing>

- [11]<https://drive.google.com/drive/folders/0B4K5KVGJNKEpOHd3cUZRRnppSDQ?usp=sharing>
- [12]<https://www.its.blldrdoc.gov/vqeg/vqeg-home.aspx>