

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。

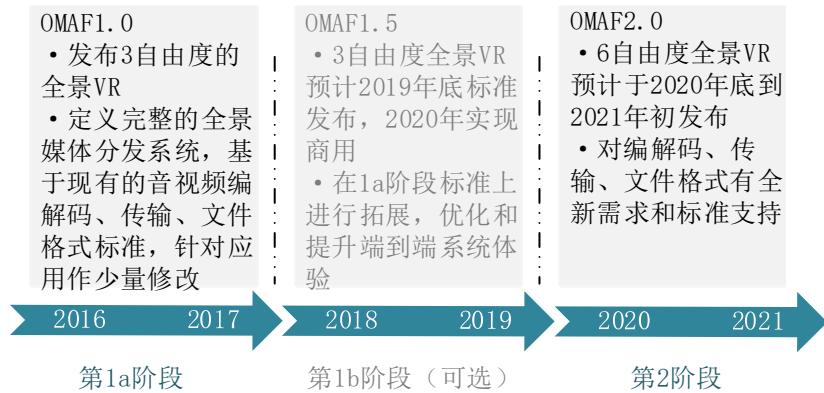


图 4.2 MPEG 组织的 OMAF 标准未来发展路线图

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。

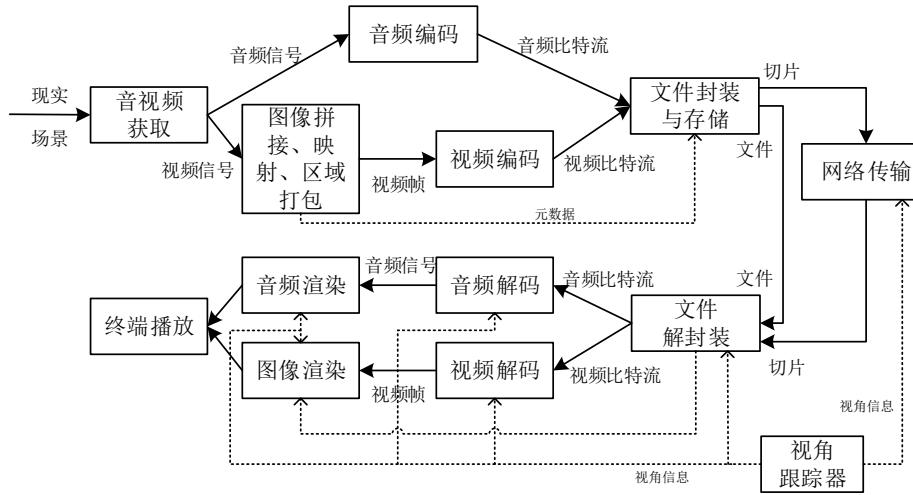


图 4.3 OMAF 下的全景媒体应用框架流程图

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 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。

4.1.2.2 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。

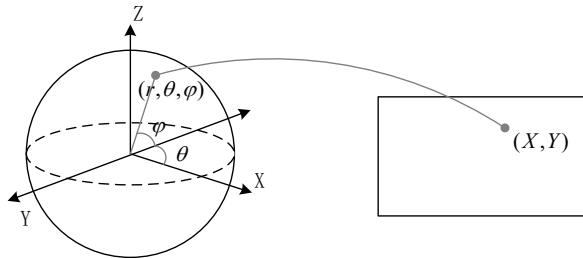


图 4.4 映射变换示意图（经纬图模型）

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。

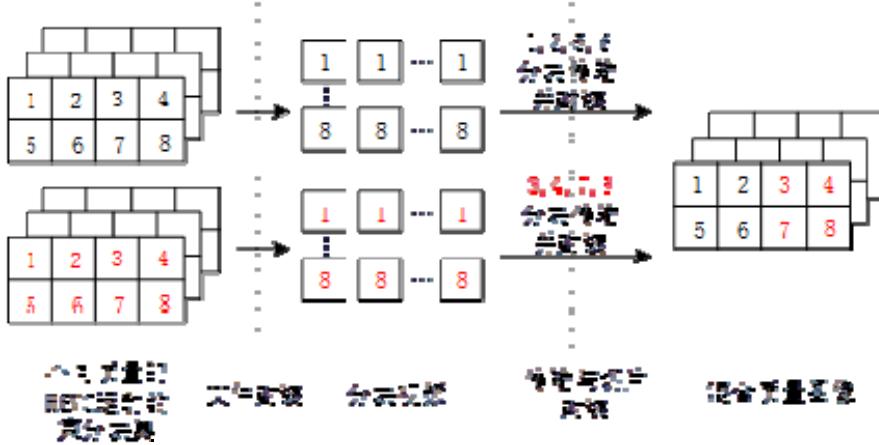


图 4.5 相同分辨率 HEVC 序列法流程图

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。

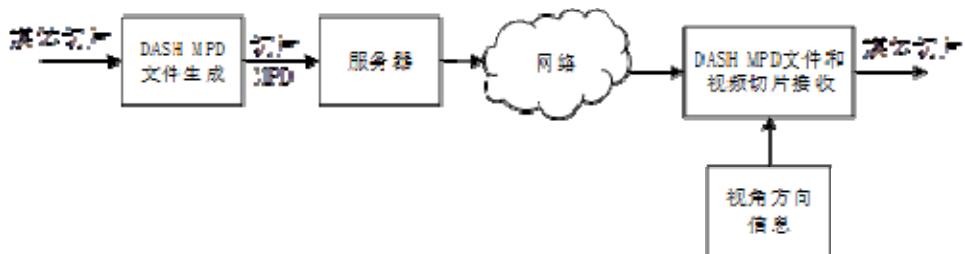


图 4.6 基于视角的 DASH 流程图

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。

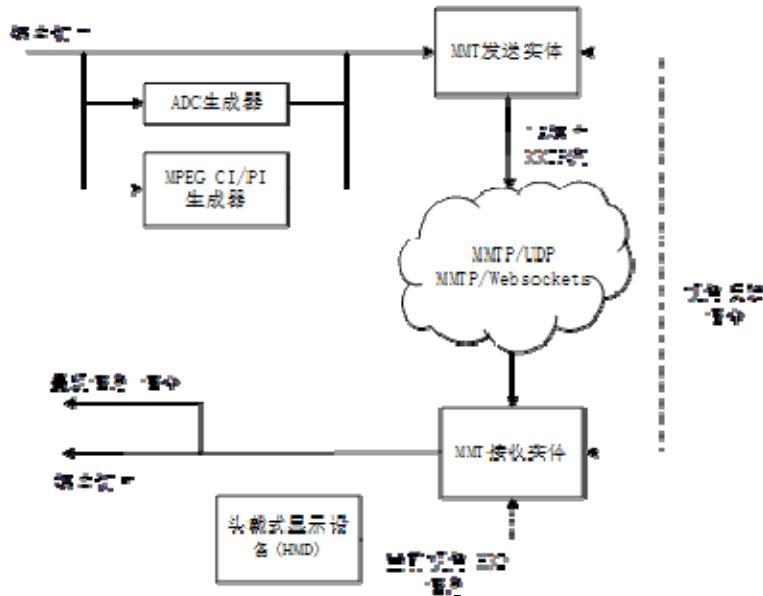


图 4.7 基于视角的 MMT 流程图

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。

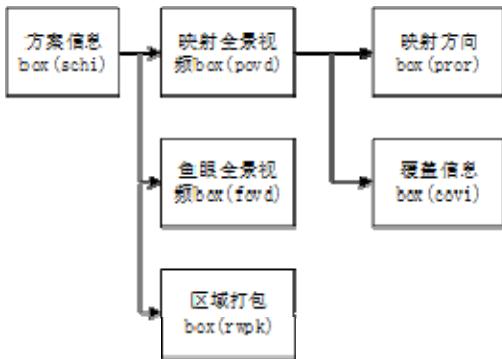


图 4.8 OMAF 在受限视频方案中的新增信息

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 (cov1) 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，它可用于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。

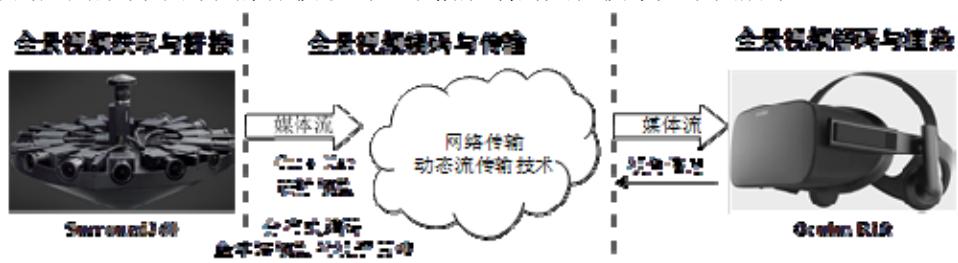


图 4.10 Facebook 的全景视频关键技术示意图

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。

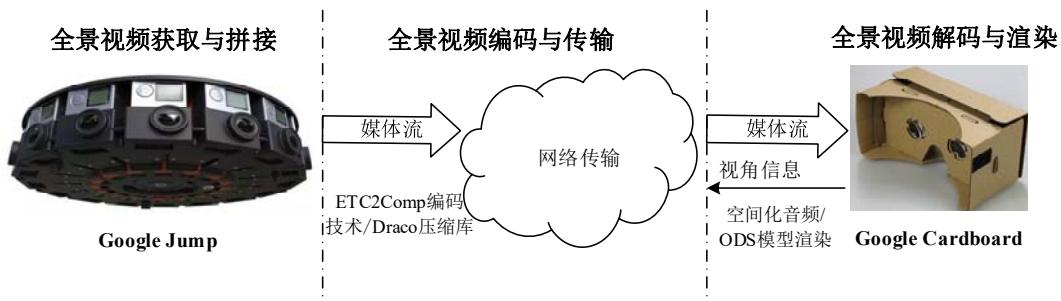


图 4.11 Google 全景媒体关键技术

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。

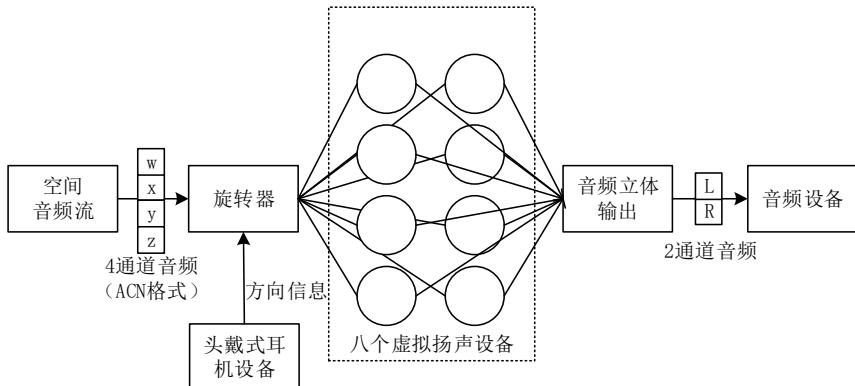


图 4.12 空间化音频技术的流程图

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。

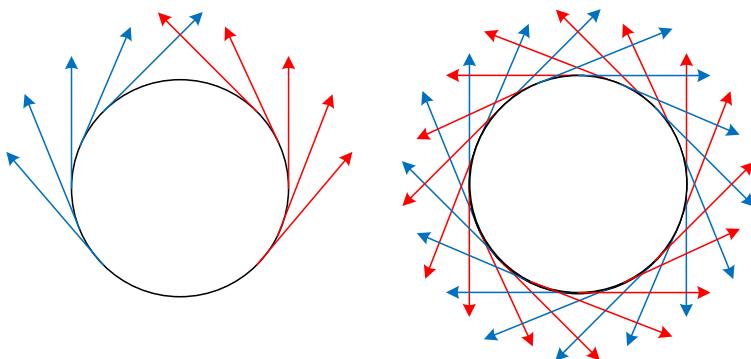


图 4.13 基于 ODS 模型的光线捕获模式

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的关键技术点，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。

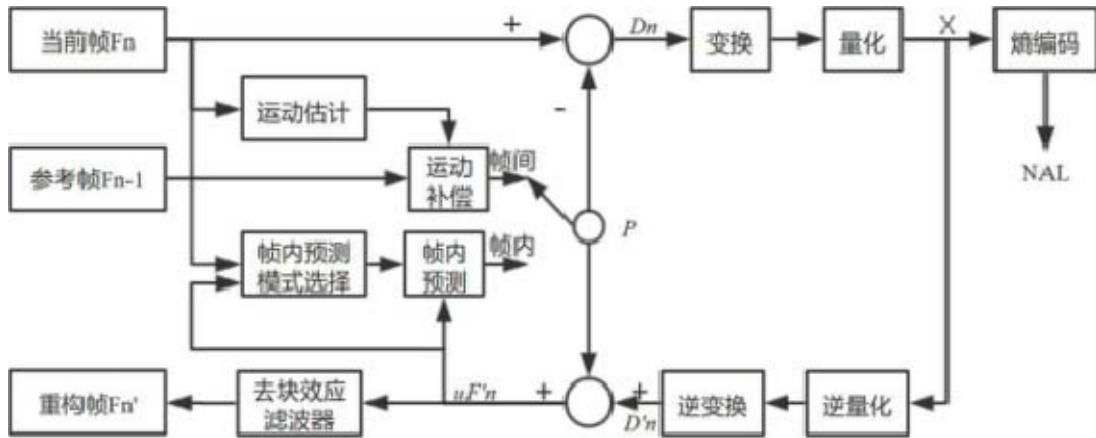


图 4.14 HEVC 编码框架

- 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

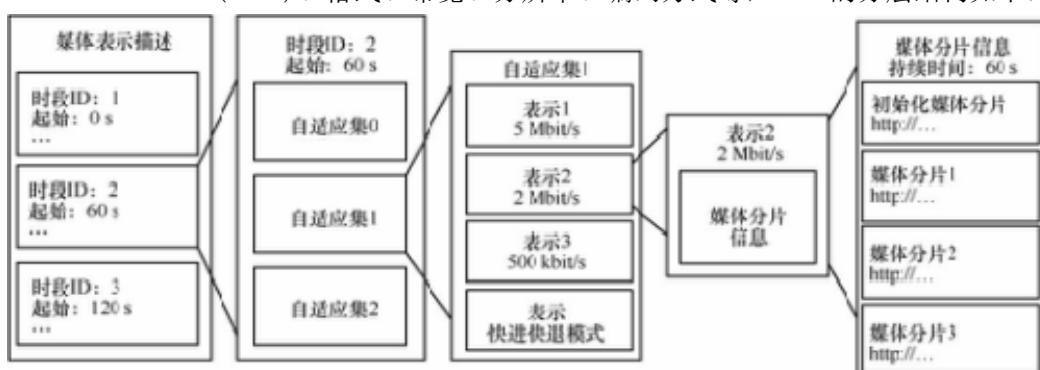
式，是 Adobe 给 RTMP 协议的补充，为 Adobe Flash 以及 AIR 客户端提供了基于 HTTP 协议的动态码流切换功能。同时它在视频编码方面支持 H.264 和 VP6，音频编码方面支持 AAC 和 MP3。协议的实现和应用依托于 Adobe 自家的 FMS (Flash Media Server)，具有动态缓存和流加密功能。流媒体索引为 manifest 文件，点播的时候通过 File Packager 将多媒体文件分割并写入 f4f 文件，这种文件格式允许通过 HTTP 协议定位内部分片并进行下载。直播的时候则是通过 Live Packager 实时收集 RTMP 流并将其转化为 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：



- 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。

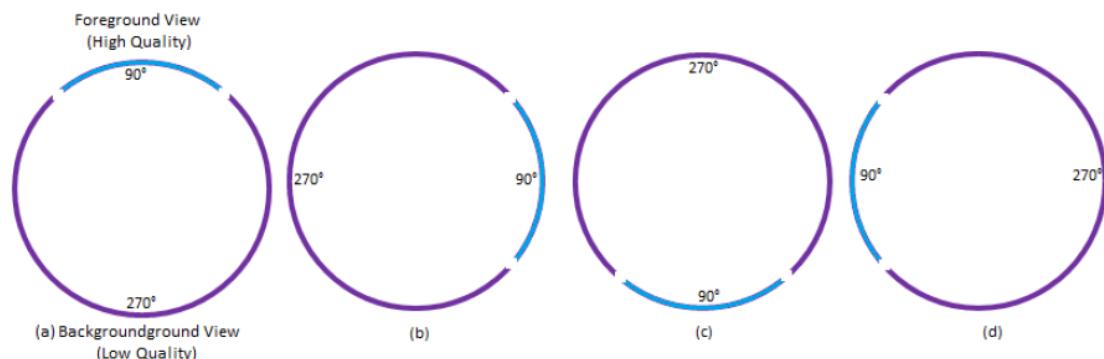


图 4.16 90 度 FoV 的 VDD

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。

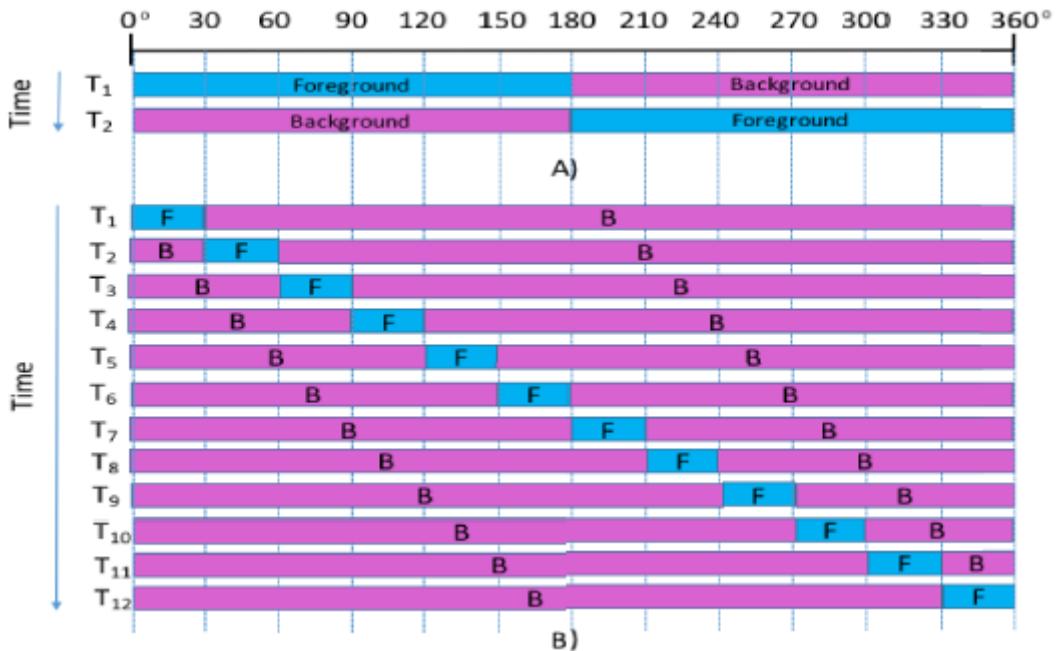


图 4.17 两种情景: A) 大视角情况 B) 小视角情况 (F 表示视角内的范围, B 表示视角外的范围)

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。

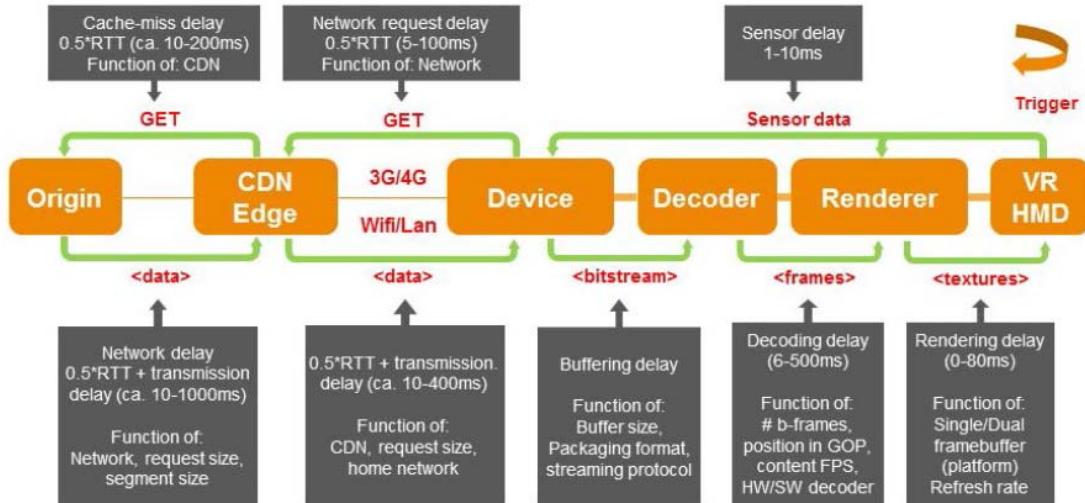


图 4.18 VR 系统中的延迟因素

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。

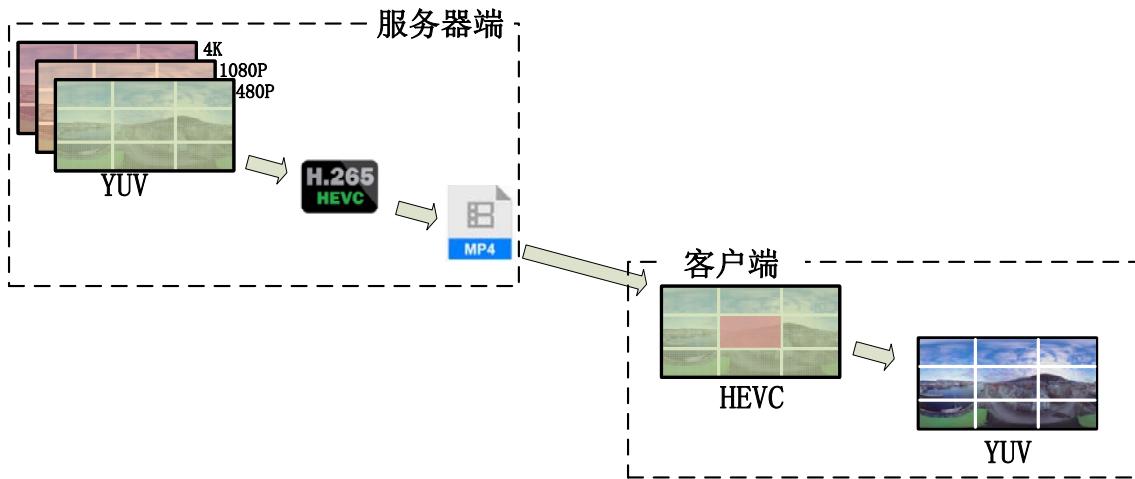


图 4.19 多分辨率分块传输系统

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。

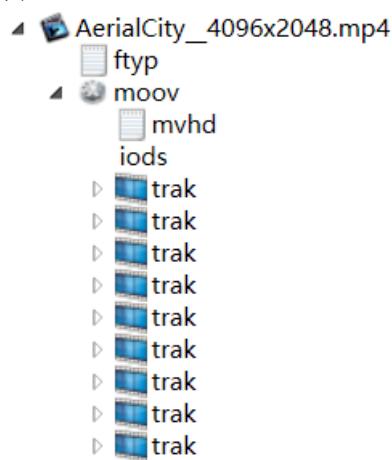


图 4.21 多通道封装

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。

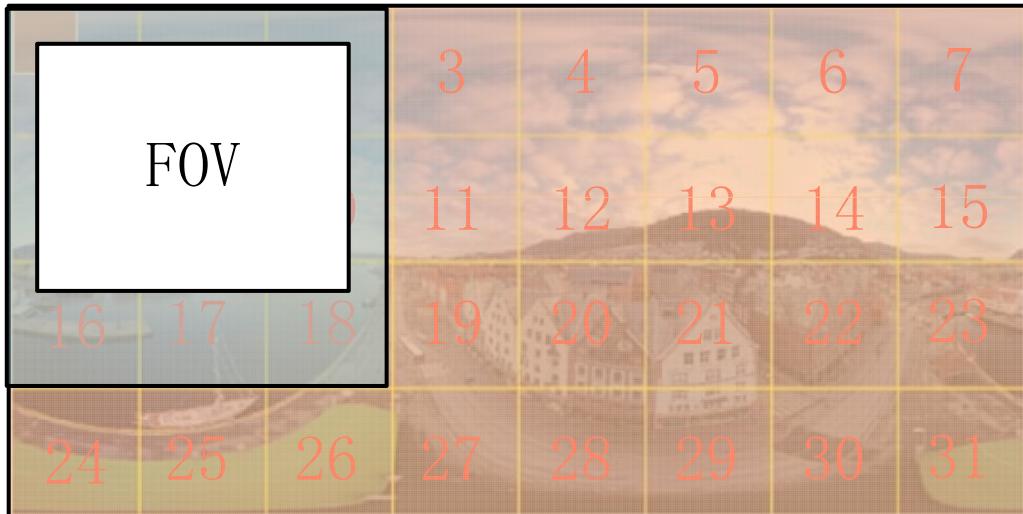


图 4.24 FoV 区域与 tile 的映射关系

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。

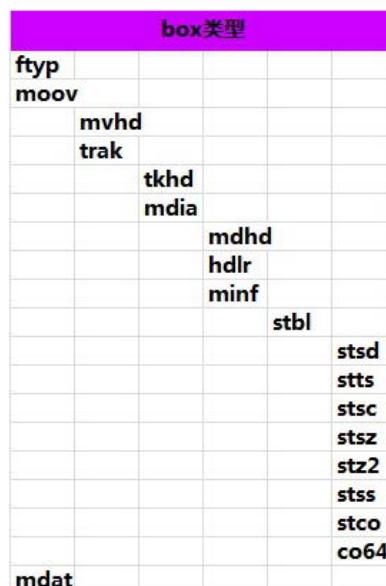


图 4.25 MP4 文件结构

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 mdat 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。

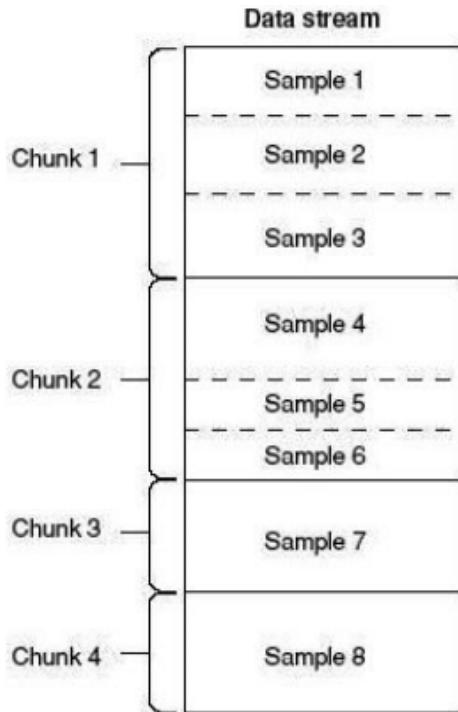


图 4.26 chunk 与 sample 的映射关系

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。

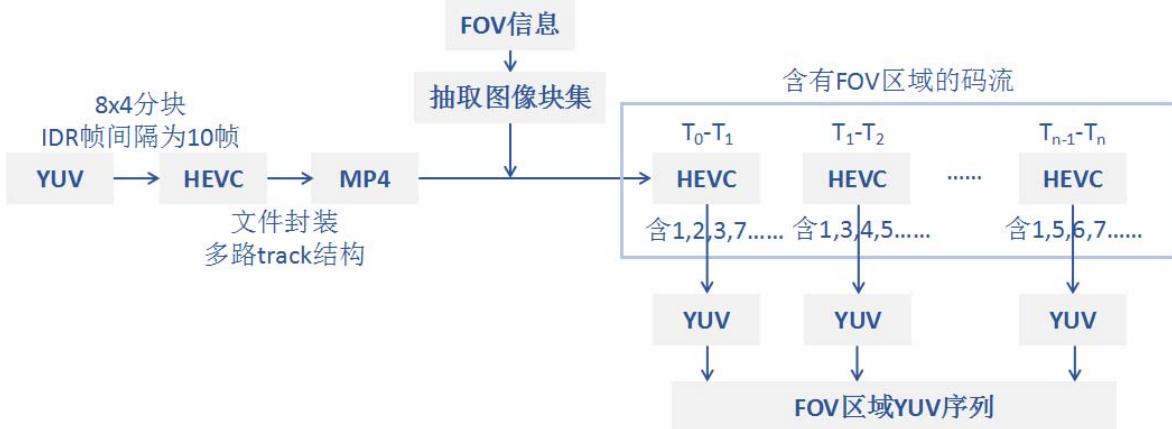


图 4.27 系统动态解码流程

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。

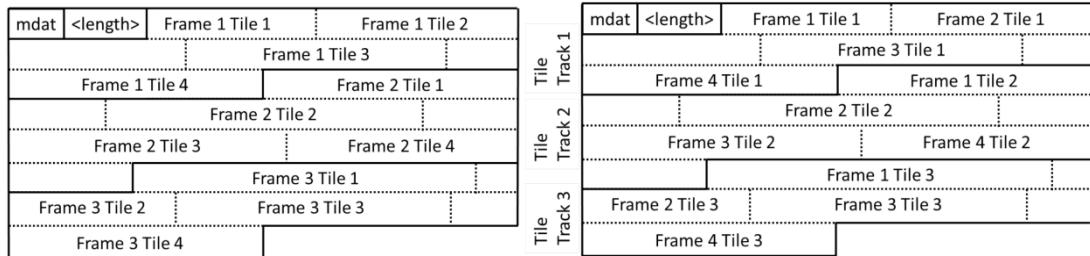


图 4.29 单路视频 track 和多路视频 track 文件的存储比较

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

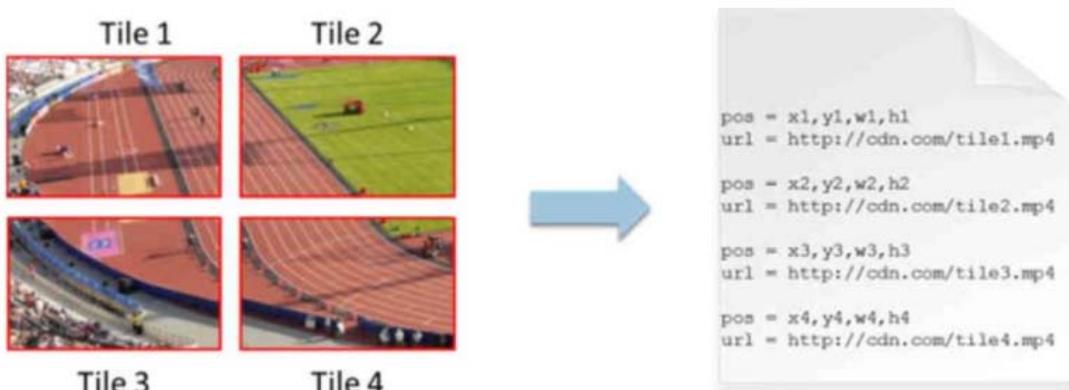


图 4.31 SRD 文件

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：

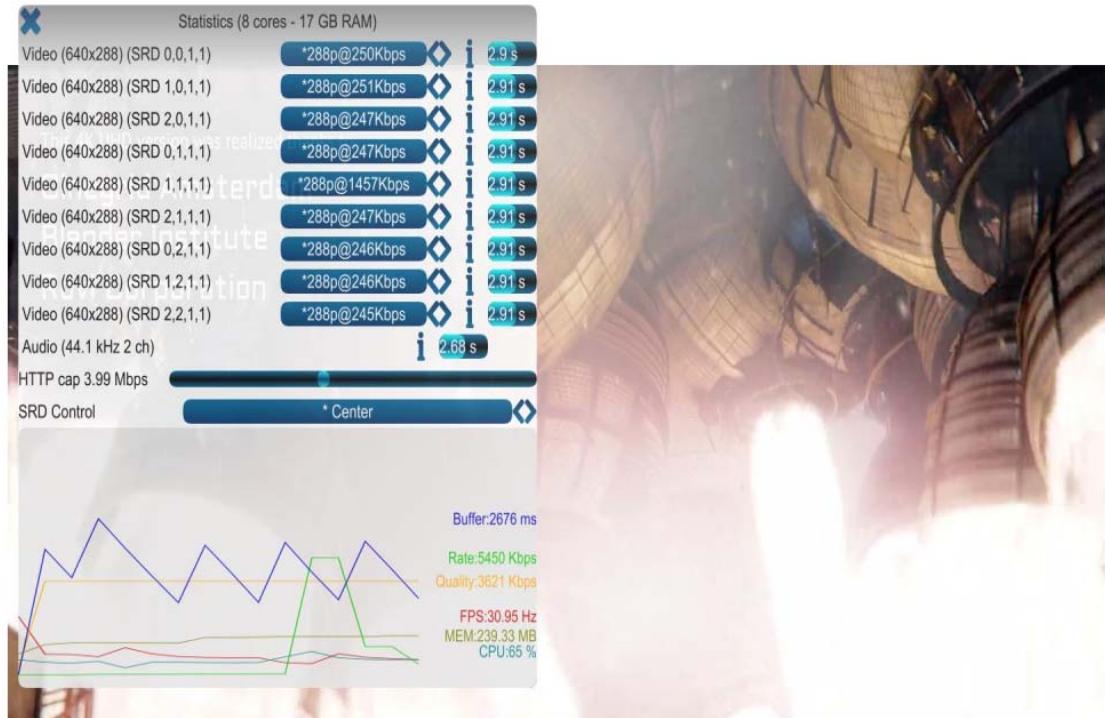


图 4.32 带有 GUI 的播放器 MP4 Client

结合上述内容，GPAC 基于 HEVC 标准的自适应传输流程如下所示：

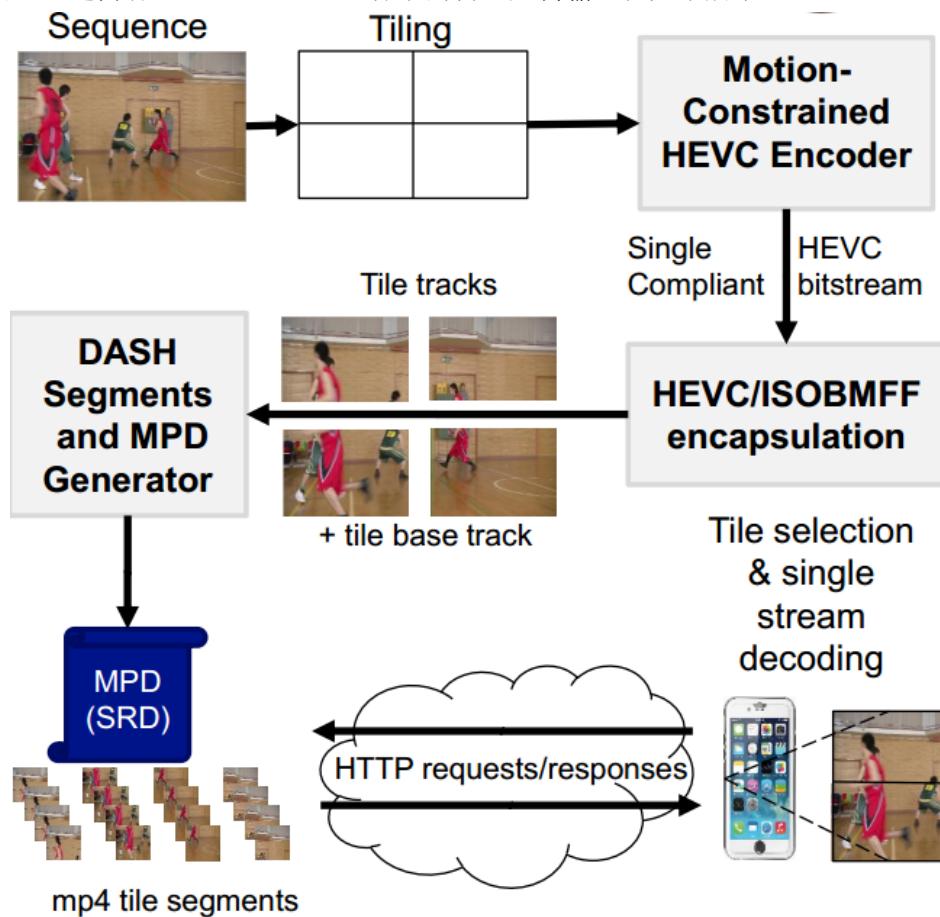


图 4.33 GPAC 的自适应传输流程

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。

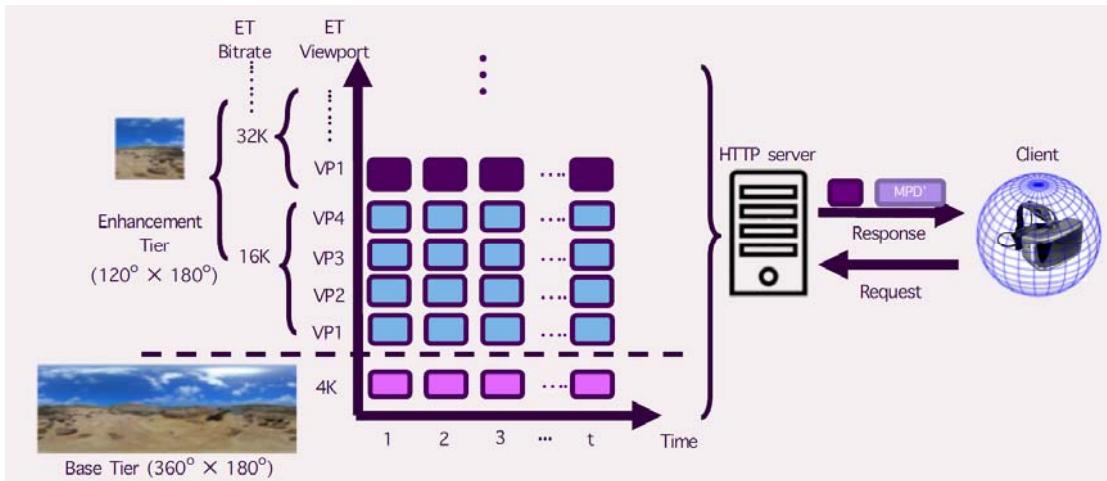


图 4.34 TTS 系统

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

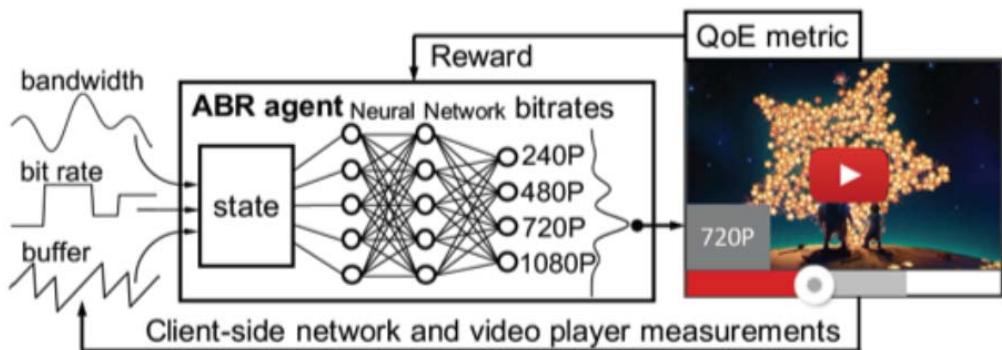


图 4.35 基于强化学习的传输决策

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：

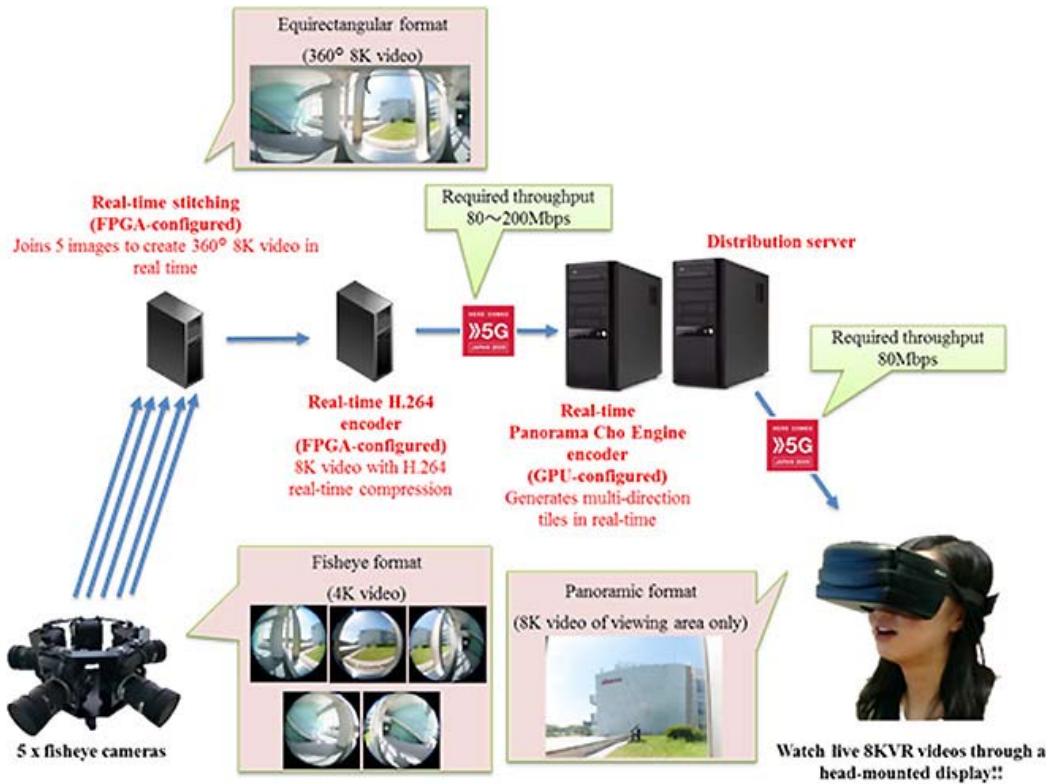


图 4.36 8K 超高清 VR 实时视频流式传输及观看系统

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

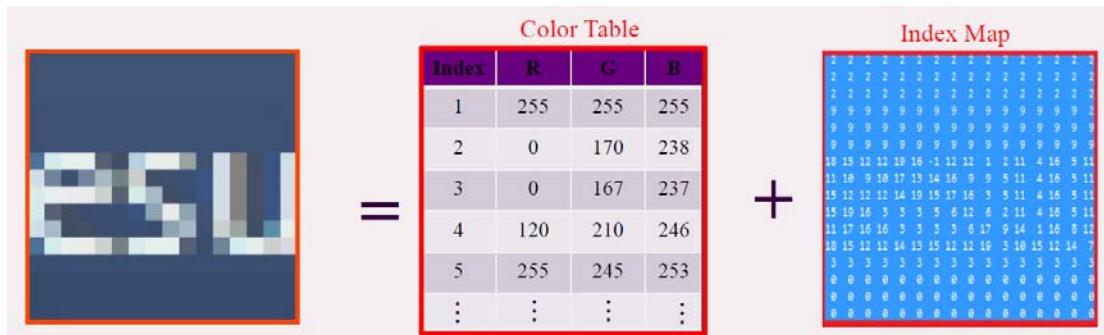


图 4.38 PLT 模式

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。

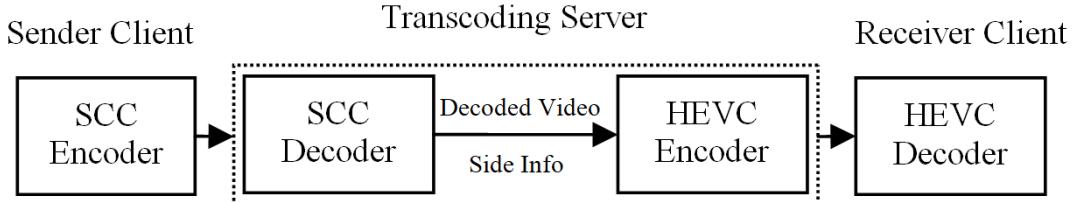


图 4.39 快速 SCC-HEVC 转码

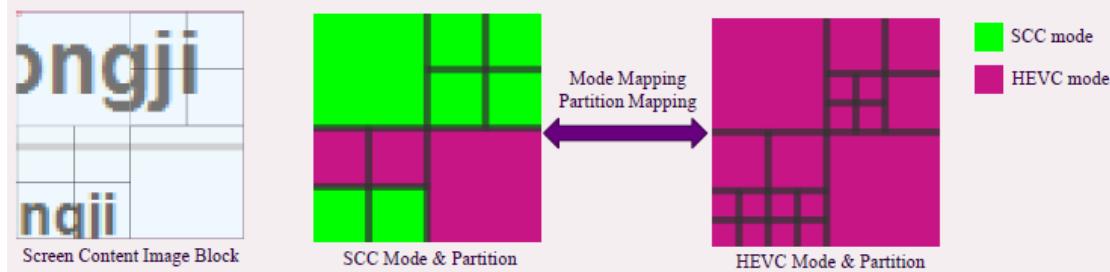


图 4.40 编码块分类与分割

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。

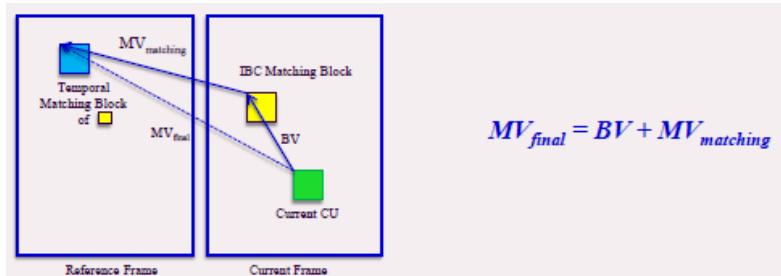


图 4.41 帧间 IBC 模式转码

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。

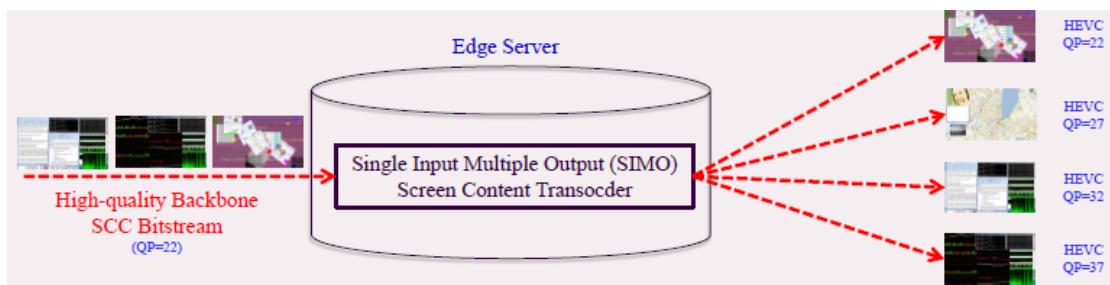


图 4.42 SIMO 转码网络

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。

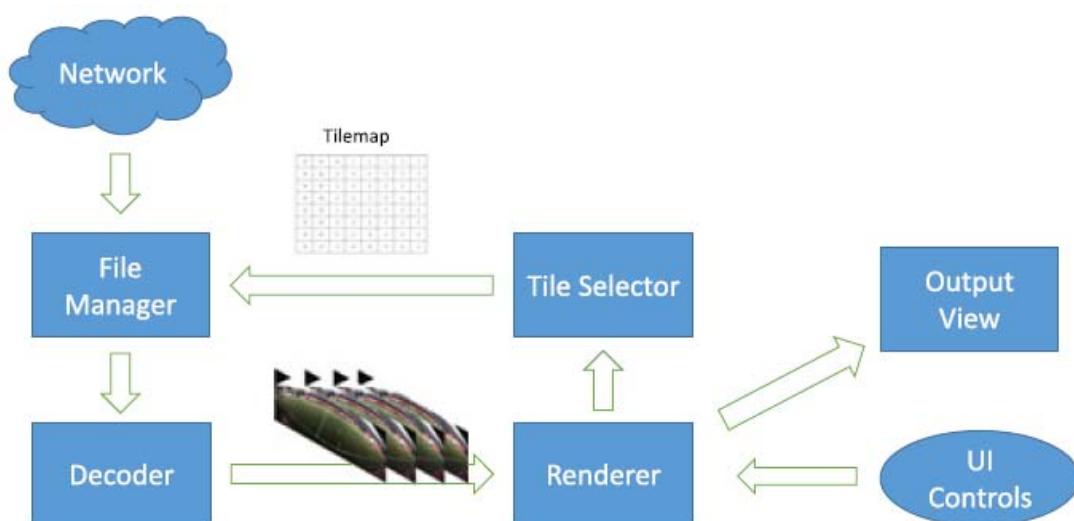


图 4.52 基于 tile 的客户端架构，其中包含了 tile 选择器

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，接下来讲简要概述一些 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。

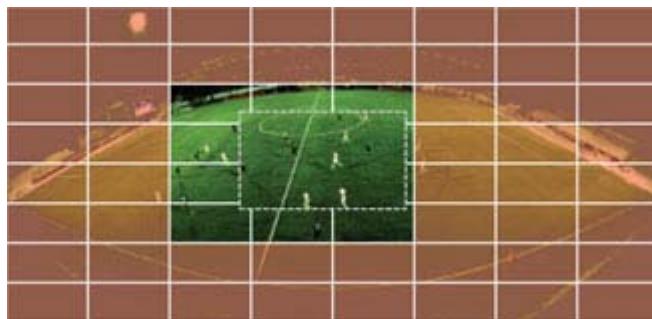


图 4.53 二元法示意图

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。

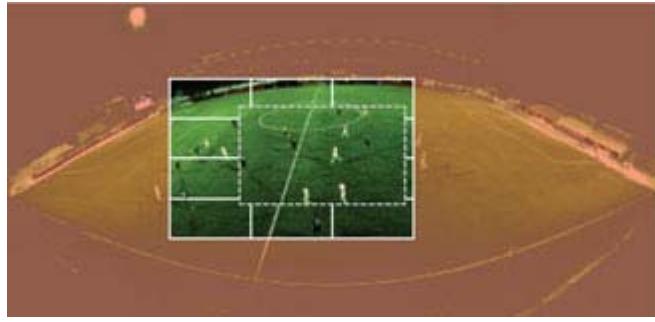


图 4.54 缩放法示意图

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。

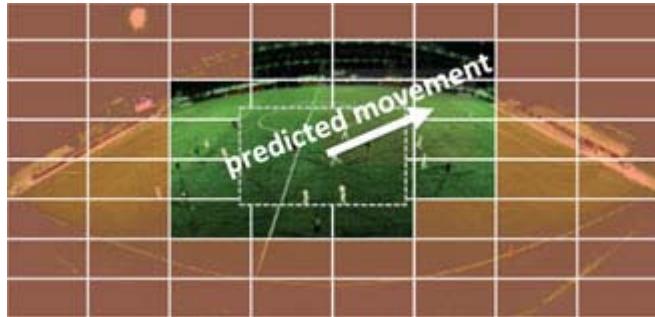


图 4.55 FoV 预测法示意图

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。

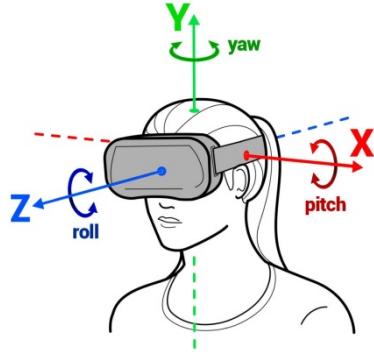


图 4.56 欧拉角头部运动模型

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_w} 、 Y_{i-I_w} 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 。

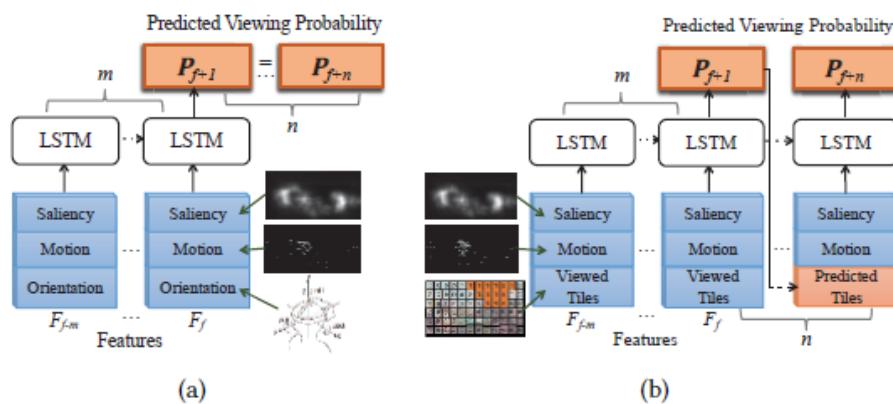


图 4.57 基于 LSTM 的 FoV 预测

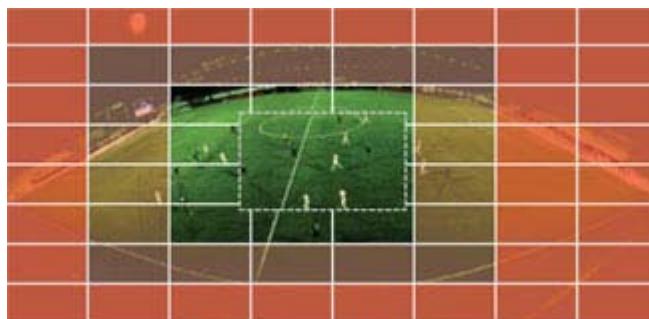


图 4.58 “金字塔”法示意图

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。

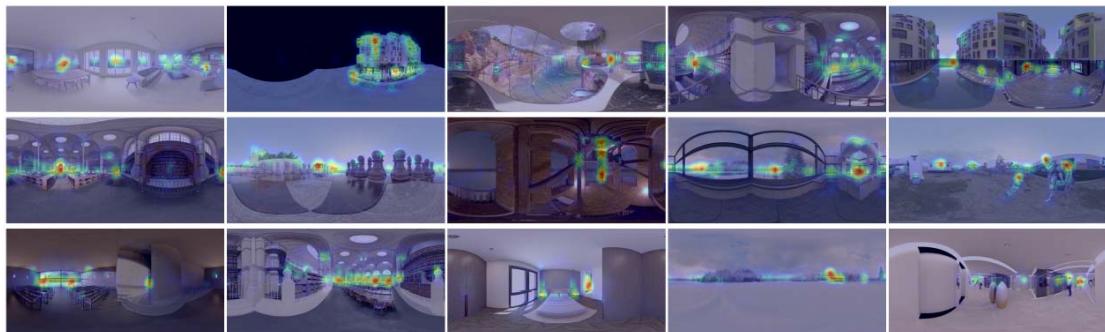


图 4.60 部分测试视频

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。

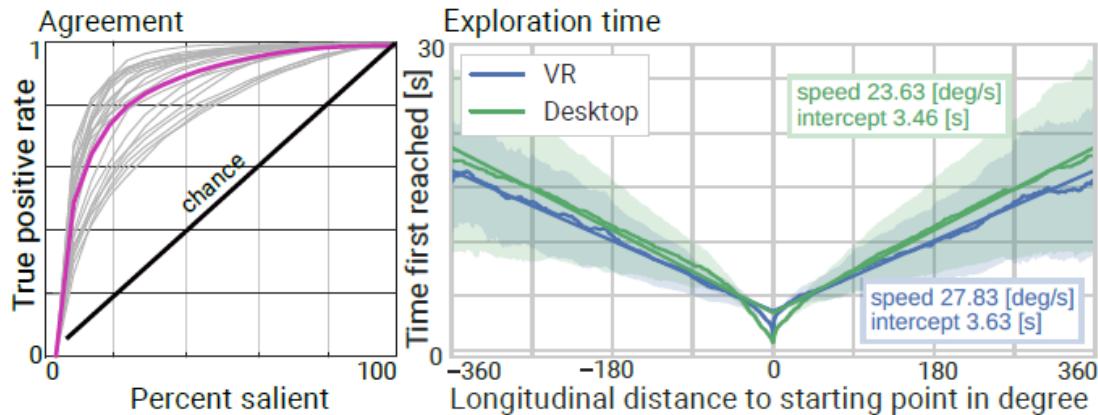


图 4.61 左：平均（红色）和各场景（浅灰色）的人类行为 ROC 曲线。快速收敛到最大值表明用户行为之间的强烈一致性。右：Exploration Time 表示与起始点达到特定经度距离时的平均时间。

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。

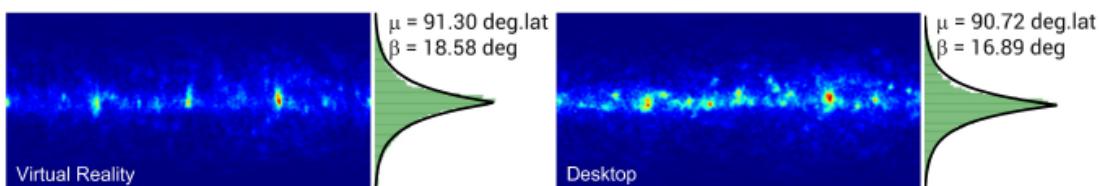


图 4.62 VR（左）和桌面（右）条件下所有场景的平均显著图



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。

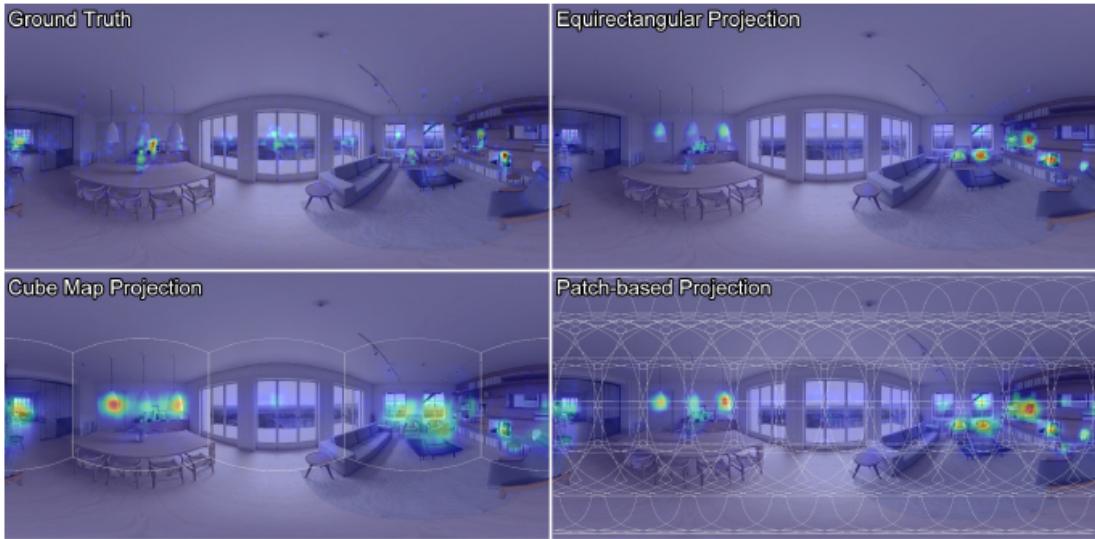


图 4.65 使用不同投影方法的显著性预测效果比较。在加入赤道偏差后，所有三种投影方法都产生了与示例相当的显著图。

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	SalNet+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 SalNet。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。

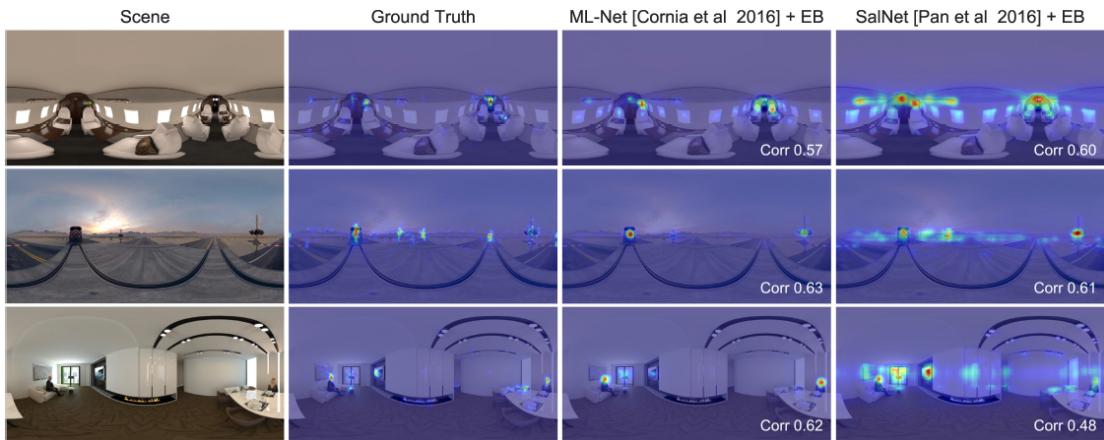


图 4.66 全景图显著性预测，这里使用基于小块的方法完成投影。通过将预测显著图与前一部分中导出的纵向赤道偏差 (EB) 相乘，团队实现了实际 (中间左侧) 和预测显著性 (右侧) 之间的良好匹配。需要注意的是，此过程可应用于任何预测方法。

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。

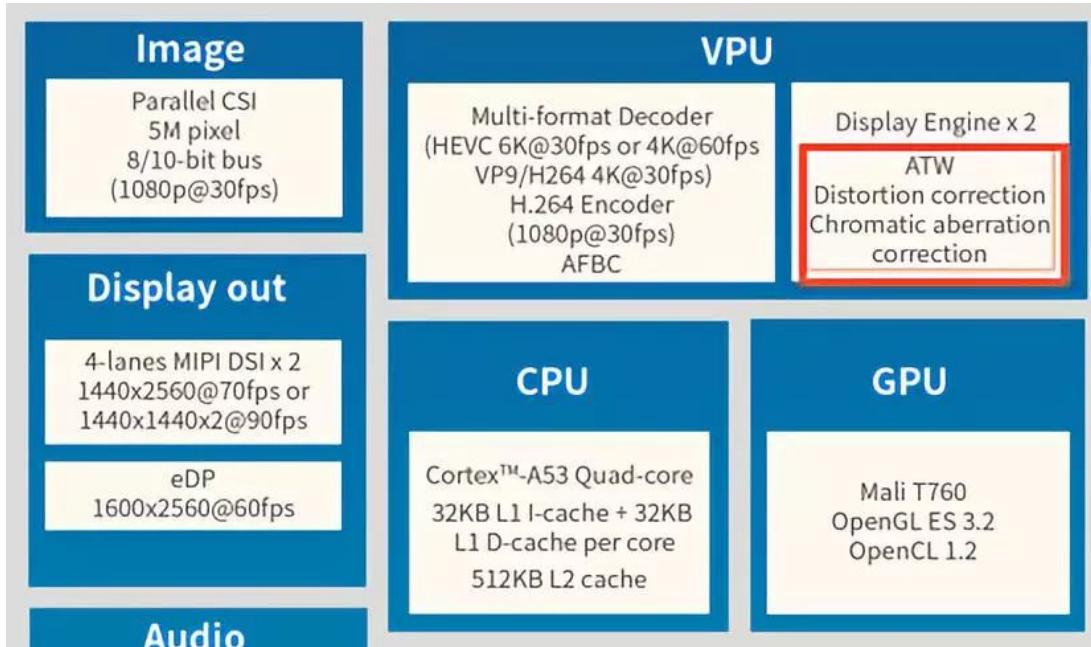


图 4.67 全志 VR9 框架图

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。

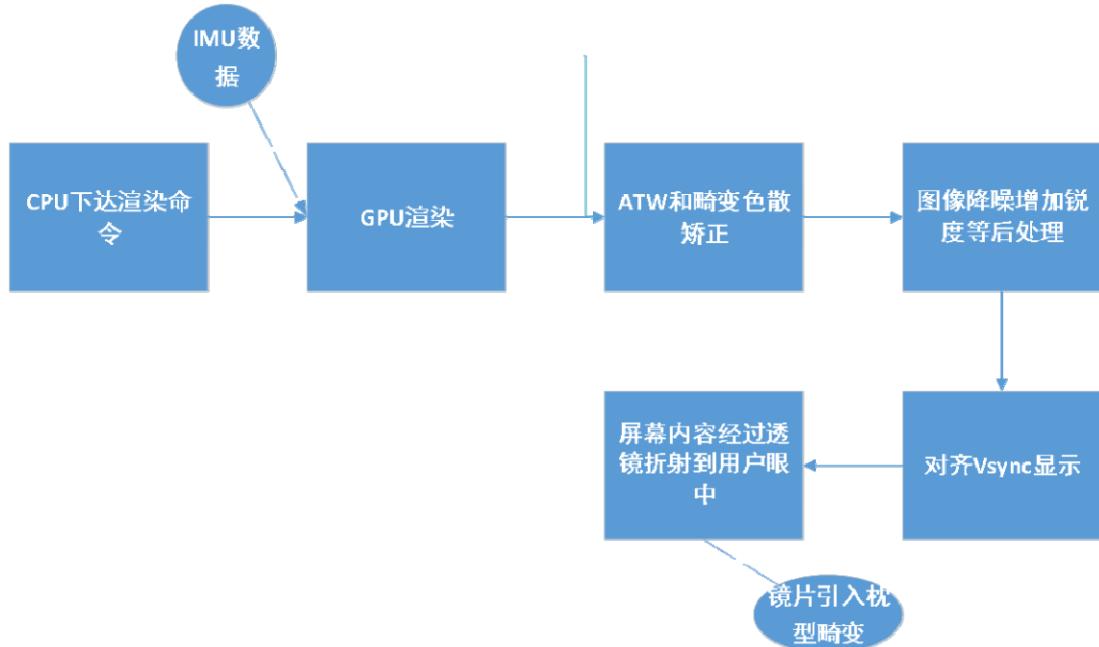


图 4.68 VR 物体进入用户眼中的历程（流水线）

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。

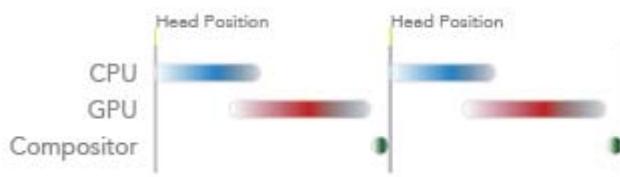


图 4.69 不带 Queue Ahead 的渲染

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。

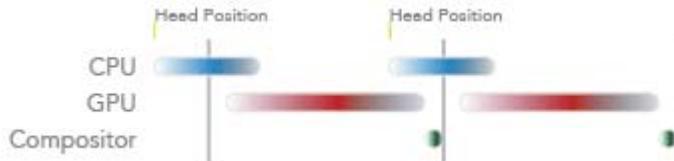


图 4.70 带 Queue Ahead 之后的渲染

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）。

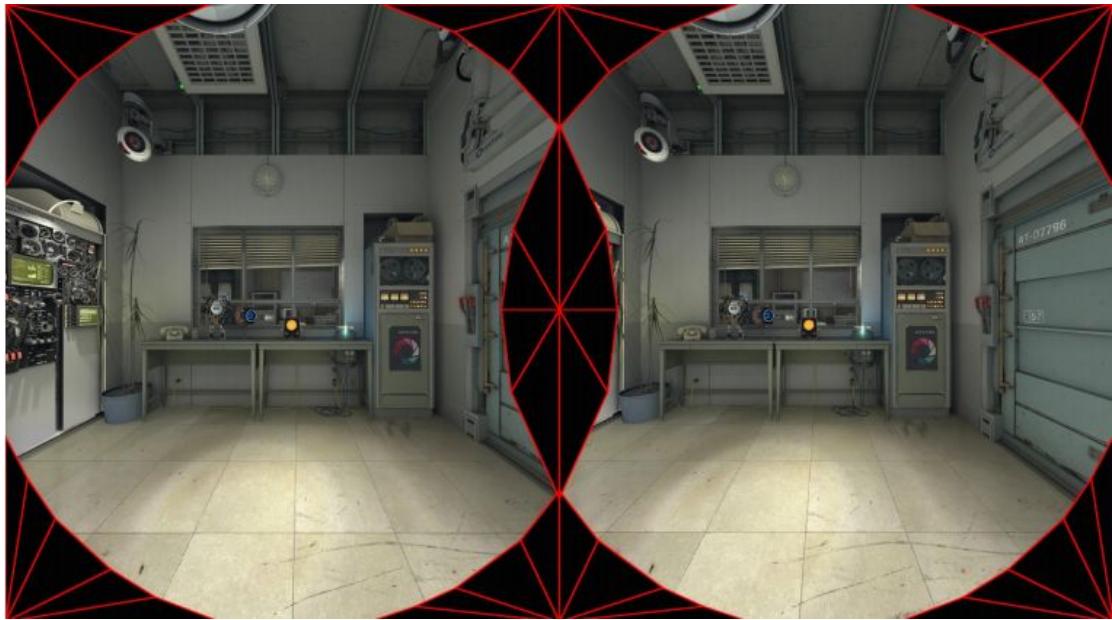


图 4.71 Hidden Mesh 技术

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杯并不能看到。



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。

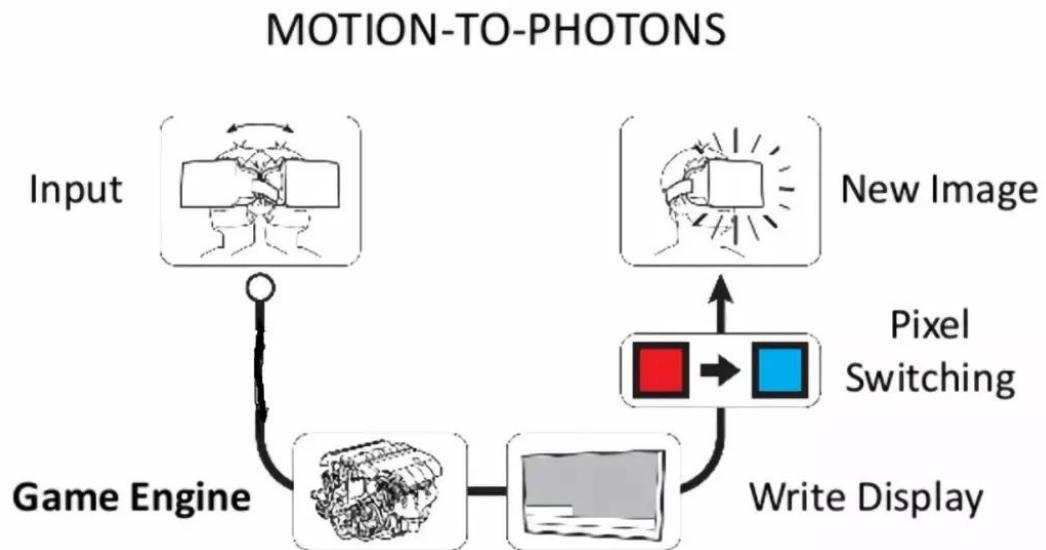
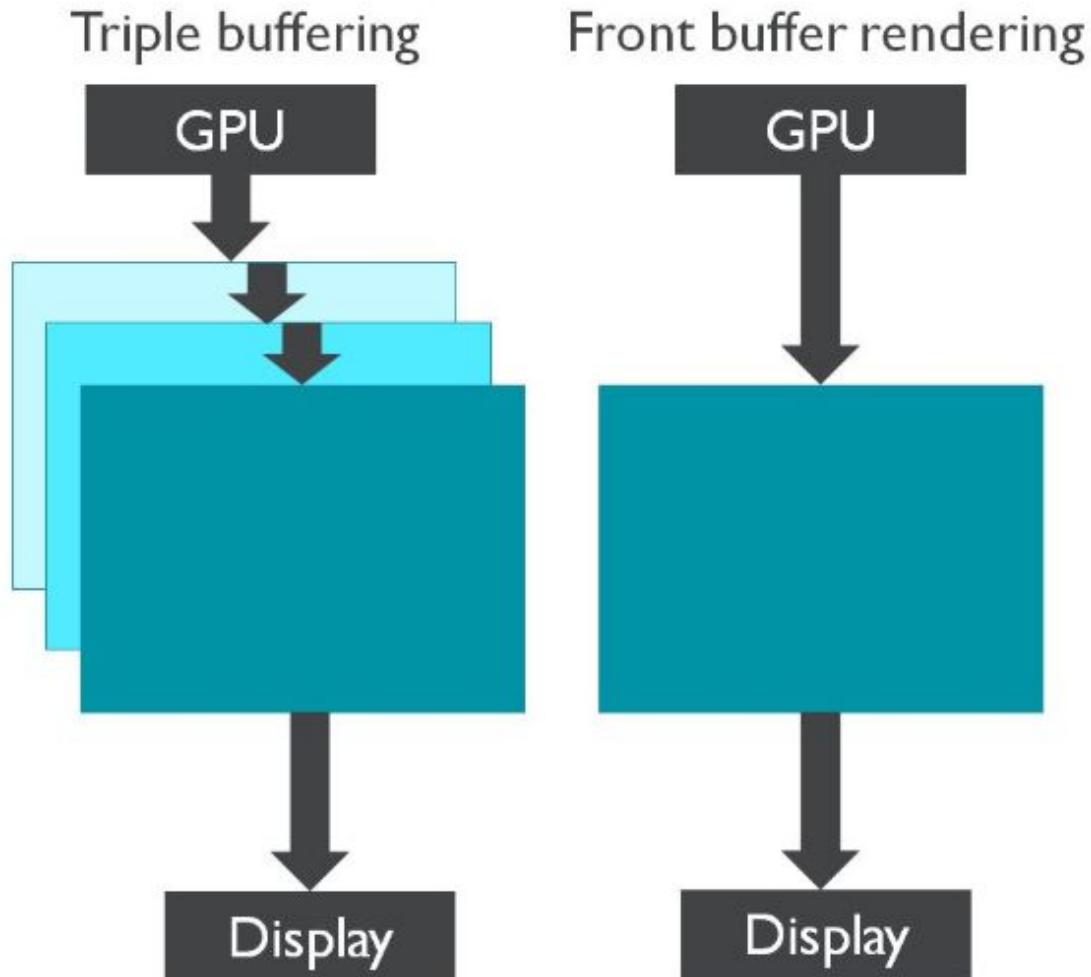


图 4.73 MTP 延迟

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。

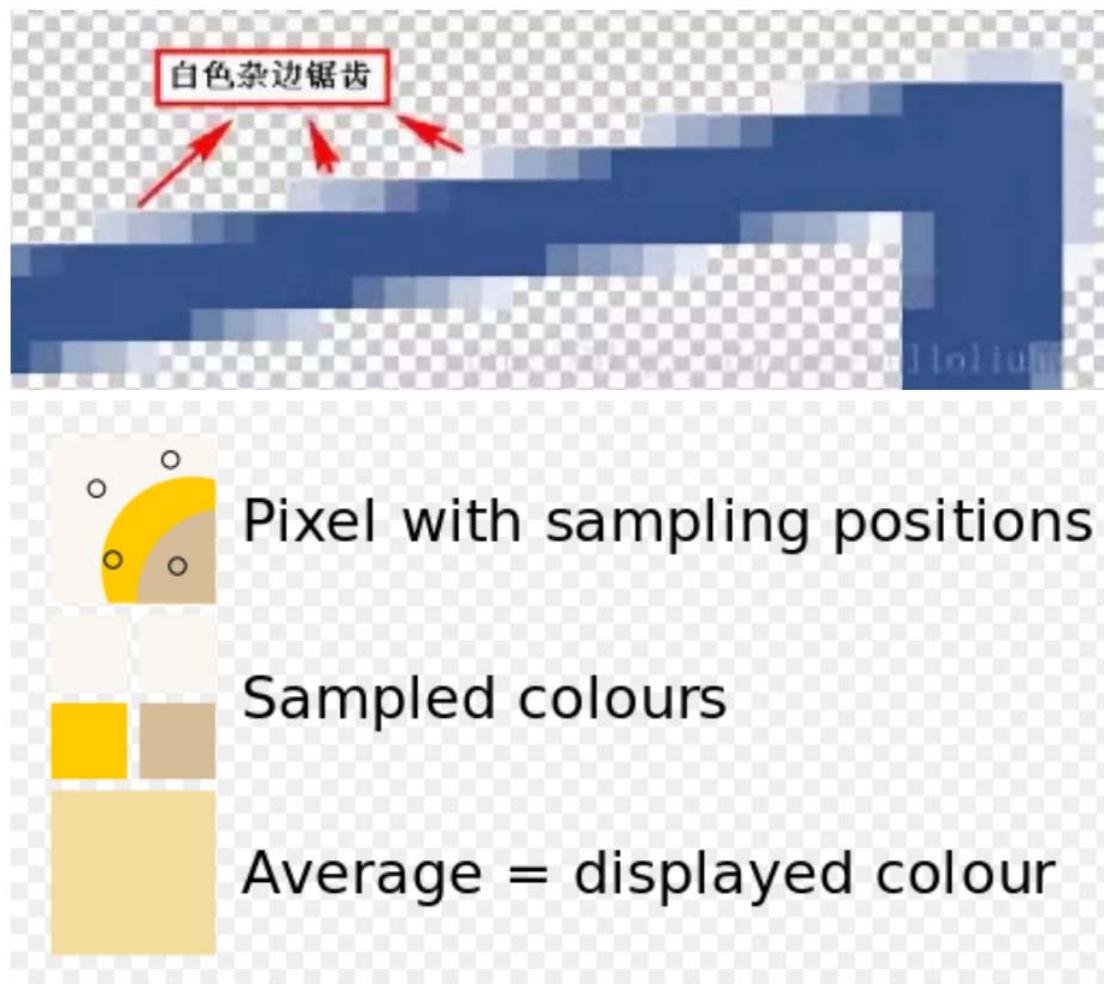


图 4.75 SSAA/MSAA 减轻锯齿

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？

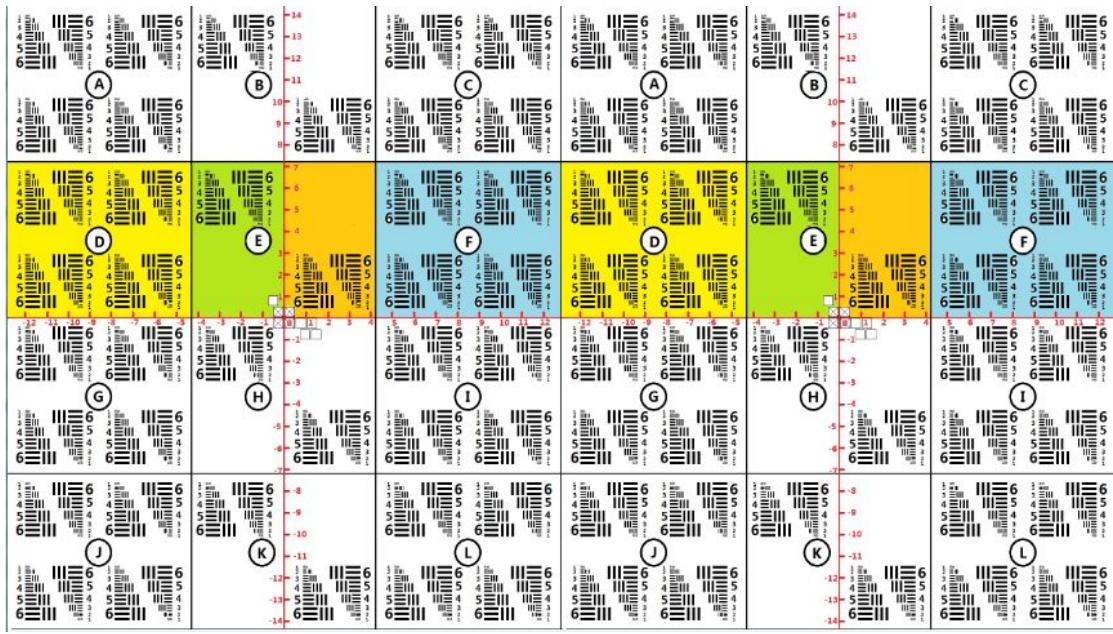


图 4.80 清晰度比较基准图

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。

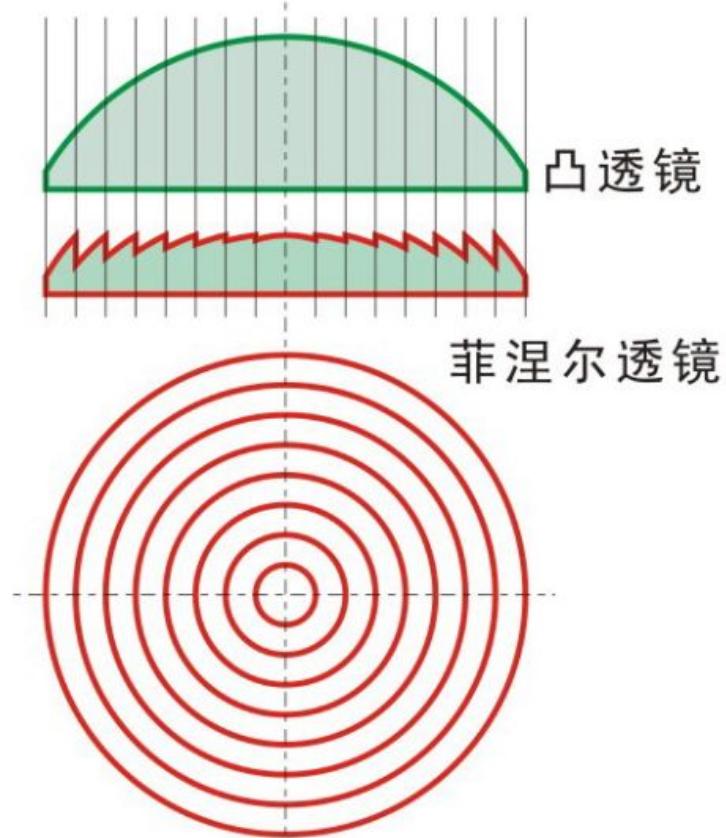


图 4.81 菲涅尔透镜外观

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。

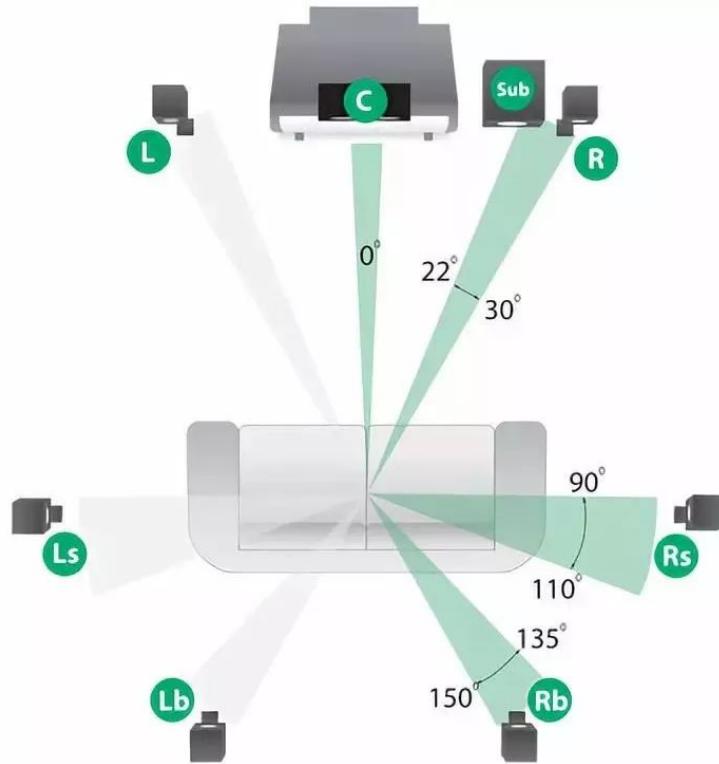


图 4.84 VR 7.1 声道

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/H2BHUKHI7ZsJEZS30ubKg>

- [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/4TmcBLXspzTeHUD0r6li0A>