



单位代码 10006

学 号 18373584

分 类 号 TP311

北京航空航天大学
B E I H A N G U N I V E R S I T Y

毕业设计(论文)

面向 360 度视频分发的多路径传输机制与实验

院（系）名称 计算机学院

专 业 名 称 计算机科学与技术

学 生 姓 名 甘天淳

指 导 教 师 张 珊

2022 年 05 月

面向360度视频分发的多路径传输机制与实验

甘天淳

北京航空航天大学

北京航空航天大学

本科毕业设计（论文）任务书

I、毕业设计（论文）题目：

面向 360 度视频分发的多路径传输机制与实验

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

本课题的研究目标是设计一种适用于 360 度视频的基于多路径的传输调度机制，并在实验室现有网络平台上进行实现和验证。在多路径传输机制设计阶段，需充分利用 360 度视频分发应用的特点，对其使用合适的编码方式，并提出视频请求与应答过程以及相应的路径调度策略。在多路径传输系统实现阶段，需基于 5G 白盒基站、Wi-Fi 接入点等设备搭建相应的物理链路，以实现所提出的多路径机制，满足用户在客户端观看 360 度视频的功能需求。

III、毕业设计（论文）工作内容：

对 360 度视频分发应用的大流量特点，设计基于多路径的传输调度机制，并在实验室现有网络平台上进行实现和验证。具体内容包括：

1. 针对 360 度视频分层分块的特征，基于网络多路径的状态设计传输调度机制，以提升终端用户的观看体验；
 2. 基于实验室现有 5G 白盒基站、WiFi 接入点等设备，实现所提出机制；
 3. 对所提方法的性能进行测试，如视频流畅度、有效吞吐率等。
-

IV、主要参考资料：

- [1] Sun L, Duanmu F, Liu Y, et al. Multi-path multi-tier 360-degree video streaming in 5G networks[A]. Proceedings of the 9th ACM Multimedia Systems Conference[C]. 2018: 162-173.
- [2] Lindskog E, Carlsson N. REEFT-360: Real-time Emulation and Evaluation Framework for Tile-based 360 Streaming under Time-varying Conditions[A]. Proceedings of the 12th ACM Multimedia Systems Conference[C]. 2021: 307-313.
- [3] Almquist M, Almquist V, Krishnamoorthi V, et al. The prefetch aggressiveness tradeoff in 360 video streaming[A]. Proceedings of the 9th ACM Multimedia Systems Conference[C]. 2018: 258-269.
- [4] Hou X, Dey S, Zhang J, et al. Predictive adaptive streaming to enable mobile 360-degree and VR experiences[J]. IEEE Transactions on Multimedia, 2020, 23: 716-731.
- [5] Zheng Z, Ma Y, Liu Y, et al. Xlink: Qoe-driven multi-path quic transport in large-scale video services[A]. Proceedings of the 2021 ACM SIGCOMM 2021 Conference[C]. 2021: 418-432.

计算机 学院 计算机科学与技术 专业类 180612 班
学生 甘天淳

毕业设计（论文）时间： 2022 年 03 月 20 日至 2022 年 05 月 25 日

答辩时间： 2022 年 06 月 07 日

成 绩： _____

指导教师： 张 珊

兼职教师或答疑教师（并指出所负责部分）：

_____系（教研室） 主任（签字）： _____

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：甘天淳

签字：

时间：2022 年 05 月



面向 360 度视频分发的多路径传输机制与实验

学 生：甘天淳

指导老师：张 珊

摘 要

360 度视频是一种新的球形视频形式，目前已经被初步应用于多个领域。这一视频形式使得用户在观看过程中的各个方向上都能产生一种深度感，从而具有良好的实用前景。但是，目前在线视频网站所支持的 360 度视频分辨率仍然较低，这是因为实时传输整个高分辨率的 360 度视频需要极高的网络带宽，限制了其发展与应用。针对这一问题，本文提出了一种基于可伸缩视频编码的多路径传输机制。

本文所提出的方法首先使用可伸缩视频编码将预先分块分段的 360 度视频分成不同质量的基础层与若干增强层，以文件的形式作为静态资源存放于服务器上。在传输过程中，利用 Wi-Fi 链路的稳定性传输各基础层以保障视频基本流畅度，利用 5G 链路的高带宽特性传输各增强层以提高用户观看的视频质量。在提出这一多路径传输机制的基础上，本文根据相关应用场景在物理设备上进行了实现。硬件层面上，搭建了 Wi-Fi 与 5G 两条物理链路并通过 IP 网段划分与静态路由配置等过程实现了全网互联互通。软件层面上，分别在多网卡服务器和单网卡 PC 上实现了基于 Django 的服务端和基于 Python 的客户端，从而基于 HTTP 协议实现了同时使用两条物理链路带宽进行 360 度视频传输的过程。

本文所实现的实机系统在功能上能够完成用户在客户端观看服务端上静态资源形式 360 度视频的需求。同时性能实验表明，在相同的网络带宽条件下，用户观看使用本文所提出多路径机制进行传输的 360 度视频过程中的卡顿频率相较于传统方法最多降低了 80%，而用于传输用户所真实观看视频部分的有效网络流量最多约是传统方法的 3 倍。这说明本文所提出的多路径传输机制能够显著提高用户体验质量。

关键词：360 度视频，可伸缩视频编码，多路径传输，5G



Multipath Transmission Mechanism and Experiment for 360-Degree Video Distribution

Author: Gan Tianchun

Tutor: Zhang Shan

Abstract

360-degree video is a new spherical form of video, which has been initially applied in many fields. It gives users a sense of depth in all directions during viewing, which leads to good practical prospects. However, real-time transmission of the entire high-resolution 360-degree video requires extremely high network bandwidth, so the resolution currently supported by online video websites is still low, which limits its development and application. To solve this problem, we propose a multi-path transmission mechanism based on scalable video coding.

We first use scalable video coding to divide the pre-tiling 360-degree video into a base layer and several enhancement layers of different quality, then store them on the server as static resources in the form of files. During the transmission process, we use Wi-Fi link to transmit the basic layers because of its stability to ensure the basic smoothness of the video, and we use 5G link to transmit the enhancement layers because of its high-bandwidth characteristics to improve the video quality watched by users. On the basis of proposing this multi-path transmission mechanism, we implement it on physical devices according to relevant application scenarios. At the hardware level, two physical links, Wi-Fi and 5G, are built, and the entire network is interconnected through IP network segmentation and static routing configuration. At the software level, a Django-based server and a Python-based client are implemented on a multi-NIC server and a single-NIC PC respectively, so as to realize the process of using two physical link bandwidths for 360-degree video transmission based on the HTTP protocol at the same time.



The real machine system implemented in this paper can fulfill the user's requirement of watching 360-degree video in the form of static resources on the server on the client side. At the same time, performance experiments show that under the same network bandwidth conditions, the frequency of freezing when users watch the 360-degree video transmitted using the multi-path mechanism proposed in this paper is at most 80% lower than that of the traditional method, while the effective network traffic of the portion of the video actually viewed is at most about 3 times that of the traditional method. This shows that the multi-path transmission mechanism we proposed can significantly improve the quality of user experience.

Key Words: 360-degree video, scalable video coding, multi-path transmission, 5G



目 录

1	绪论	1
1.1	研究背景与意义	1
1.2	国内外研究现状	2
1.3	研究目标与内容	3
1.4	本文组织结构	4
2	相关研究	5
2.1	360 度视频的预处理	5
2.1.1	网格式的切分方法	5
2.1.2	基于视角的切分方法	6
2.1.3	综合的切分方法	6
2.2	360 度视频的传输机制	7
3	多路径传输机制设计	10
3.1	传输机制的性能指标设计	10
3.1.1	卡顿频率	10
3.1.2	有效流量	11
3.2	360 度视频的预处理与编码	12
3.2.1	360 度视频的格式转换	13
3.2.2	360 度视频的切分	14
3.2.3	360 度视频的编码	15
3.3	360 度视频块的请求与应答过程	17
3.4	360 度视频块的路径调度策略	18
4	基于 Wi-Fi 与 5G 链路的多路径系统实现	20
4.1	系统需求分析	20
4.2	系统总体架构	21
4.3	物理链路层搭建与调试	22
4.4	应用业务层关键模块实现	23



4.4.1 360 度视频静态资源的生成	23
4.4.2 服务端	24
4.4.3 客户端	26
5 系统测试与实验结果	28
5.1 测试数据	28
5.2 系统功能展示	29
5.3 系统性能测试	29
总结与展望	32
致谢	33
参考文献	34

1 绪论

1.1 研究背景与意义

随着计算硬件和网络技术的进步以及高带宽的支持, 360 度视频和虚拟现实 (VR) 技术逐渐被初步应用于包括娱乐、教育、制造、交通、医疗在内的多个领域。360 度视频是一种可被映射成 3D 几何体的球形视频, 如图 1.1 所示, 用户可以在播放过程中使用 VR 头戴式显示器环顾四周, 这让观众在各个方向都有一种深度感。作为新一代多媒体的基础内容, 360 度视频在多领域的应用都展现出巨大的潜力, 它可以提供更丰富、更吸引人和更沉浸式的体验, 从而被企业和消费者采用。近年来, 360 度视频越来越受到公众的关注, 这是因为它们可以为用户提供即时场景, 从而提高视频点播和直播视频流的体验质量^[1]。许多大型商业视频服务提供商, 如 YouTube 和爱奇艺等, 都已提供了 360 度视频流服务。这些新兴的视频服务有助于吸引新的消费者, 具备为视频流业务带来巨大利润的潜力。根据市场研究^[2], VR 和增强现实 (AR) 系统预计将在 2025 年达到大致相当于当今台式 PC 市场的规模。

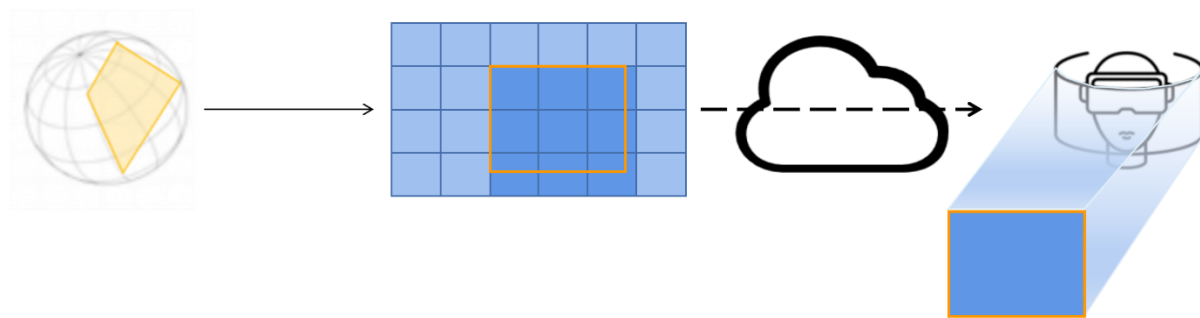


图 1.1 360 度视频的结构与用户视角

然而, 360 度视频的发展也面临着独特的新挑战。由于需要确保向用户显示视频的质量良好, 360 度视频通常需要以 4K 或更高分辨率进行播放。但在现实情况下, 通过互联网实时传输整个高分辨率的 360 度视频内容需要极高的带宽, 研究表明^[3], 一个每秒 120 帧、分辨率为 24K 的优质 360° 视频可以轻松消耗每秒数千兆比特 (Gbps) 的带宽, 这在使用中会造成卡顿、模糊等对用户不友好的情况, 因此目前经常使用先下载再观看的模式, 这使得 360 度视频难以取代传统的 2D 视频, 限制了其发展与应用。因此尽管



360 度视频具有良好的前景,但与传统 2D 视频相比,现有的 360 度视频应用程序能够支持的分辨率仍然较低,难以做到在线实时高分辨率播放,这其中最主要的挑战即是如何将庞大的 360 度视频流高效地传输到带宽受限的设备上。

基于这一问题,研究并实现一种适用于 360 度视频的传输机制便具有重大的意义。对于视频服务提供商而言,这一机制将会降低实时传输 360 度视频所需的带宽,从而使得提供更高分辨率的 360 度视频成为可能。对于用户而言,这一机制能够降低实时观看过程中的卡顿频率,并提高用户视口内的视频质量,进而优化用户体验。进一步的,该机制还对虚拟现实(VR)技术等的发展有重要意义。

1.2 国内外研究现状

基于本文的研究目标,主要调研了关于 360 度视频传输机制的相关国内外研究现状。传输高分辨率下的 360 度视频需要极高的带宽,为了减轻实时传输的带宽需求,获得更好的用户体验,目前已有许多相关研究与工作尝试对 360 度视频这一视频类型使用特殊的编码策略或传输机制。

由于 360 度视频在进行处理和传输之前首先需要投影到 2D 平面上,所以一些工作尝试使用空间上更佳的投影方式从而优化需要传输的数据量,也即使用更佳的格式对球形 360 度视频进行表示。例如 2016 年 Facebook 提出的立方体格式^[4]和金字塔格式^[5]投影方法以及其对应的编码方案,在他们的实验中分别实现报告了 25%和 80%的改进。

另一些工作尝试使用具有压缩思想的源位分配(Source Bit Allocation)操作,这是因为 360 度视频的不同区域具有不同的感知质量,从用户角度而言可以使用不同的位数进行编码。例如 Budagavi 等人提出的一种区域自适应的平滑方案^[6],他们利用高斯滤波降低了 360 度视频南北两极区域的比特数,从而使得用户感知质量下降不明显的条件下传输所需的比特率降低了 20%。

此外,近年来提出了许多基于视频块(tile)的处理与传输机制,这些机制往往为用户最感兴趣的区域分配更高的视频质量和传输优先级,而为用户不感兴趣的区域分配较低的质量与优先级。例如 Duanmu 等人提出了一种基于视图预测的流传输框架^[7],该框架通过仅对用户最终想要的视频部分进行传输从而减少对网络带宽的消耗,并在此基础上提出了一种基于视点预测精度的动态视频分块自适应算法来调整分块的覆盖率,他们

的实验表明,在不考虑编码效率损失的情况下,传输所需的下载速率最多降低了 80%。

1.3 研究目标与内容

本文将通过将 360 度视频分片分层并进行多路径传输的方式减少实时传输 360 度视频所需的网络带宽,从而降低视频卡顿频率,提高用户视口内视频质量,进而提高用户体验质量。本文将主要针对服务器上存储的已经过预处理的 360 度视频的多路径传输进行研究和实验,采用三个步骤依次完成研究与实现。首先,需要下载 360 度视频并使用可伸缩视频编码(SVC)将视频转化为静态资源存储于服务器上。其次确定多路径传输系统的系统架构并构建其整体框架,确定每一部分的设备,功能及其实现形式,并初步进行系统搭建,在该系统上依次实现普通数据流的单路径与多路径传输。进一步的,需要设计针对 360 度视频的多路径传输机制,将已切分好的 360 度视频按一定策略规划到多条路径上进行传输,并进行有关实现。主要研究内容模块如图 1.2 所示。

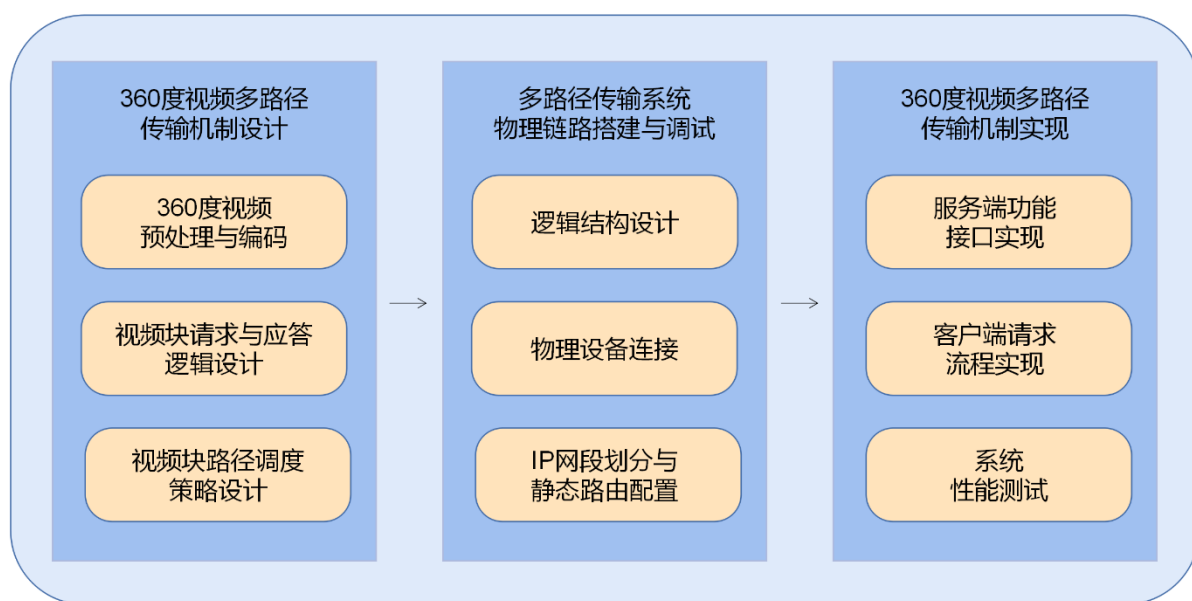


图 1.2 研究内容模块图

针对以上研究问题和研究目标,本文的工作内容可主要分为如下三点:

1) **360 度视频多路径传输机制的设计:** 本文提出了一种基于可伸缩视频编码的多路径传输机制。预处理与编码方式上,本文的方法首先将 360 度视频按一定分辨率和时长分块分段并使用可伸缩视频编码分为不同质量的一个基础层和多个增强层,形成静态资源并以文件形式作为传输的基本单元。在此基础上本文设计了一个实时的视频块请求



与应答过程,服务端通过客户端所发送的用户视点决策需要传输的视频块层文件并完成传输过程。客户端根据收到的文件解码出对应的视频并向用户播放。路径调度策略上,本文利用 Wi-Fi 链路的稳定性传输各基础层以保障视频基本流畅度,利用 5G 链路的高带宽特性传输各增强层以提高用户观看的视频质量。

2) 多路径传输系统的物理链路搭建与调试: 本文在实验室现有设备的基础上搭建起了一个多路径传输系统的物理框架,即搭建了 Wi-Fi 与 5G 两条物理链路,并通过 IP 网段划分与静态路由配置等过程实现了全网互联互通,在此基础上实现了普通数据流的多路径传输。

3) 360 度视频多路径传输机制的实现: 本文在完成物理链路搭建与调试的基础上,分别在链路两端的多网卡服务器和单网卡 PC 上实现了基于 Django 的服务端和基于 Python 的客户端,从而基于 HTTP 协议实现了同时使用两条物理链路带宽进行 360 度视频传输的过程,从功能上完成了用户在客户端观看服务端上静态资源形式 360 度视频的需求。

1.4 本文组织结构

本文的组织架构主要分为以下五个部分,每个部分的组织内容如下所示:

第 1 章: 绪论。本章首先介绍了本文的研究背景及意义,然后对所研究课题的国内外研究现状、研究目标及内容以及本文的组织结构进行了简要介绍。

第 2 章: 相关研究。本章从 360 度视频的预处理方法以及其具体传输机制这两个主要方面对目前本领域的国内外相关研究进行了简要介绍。

第 3 章: 多路径传输机制设计。本章首先介绍了传输机制的评价指标设计,并以此出发点详细介绍了本文所提出的多路径传输机制设计。

第 4 章: 基于 Wi-Fi 与 5G 链路的多路径系统实现。本章首先对 360 度视频多路径传输系统的需求进行分析,并在此基础上提出所实现系统的整体架构,随后依次对整体架构中的各个层次内的各个重要模块进行详细介绍与分析。

第 5 章: 系统测试与实验结果。本章首先介绍了实验所使用的测试数据,随后给出了系统功能测试和性能测试的相关结果,并进行了一定的对比分析。

最后,对本文的工作进行了一定总结,并对接下来的工作提出展望。



2 相关研究

传输高分辨率下的 360 度视频需要极高的网络带宽,为了减轻实时传输的带宽需求,获得更好的用户体验,目前已有许多相关研究与工作尝试对 360 度视频这一视频类型使用特殊的传输策略,而为了使得这些传输策略更加有效,这些工作往往需要对视频进行合适的预处理。下面将依次介绍一些主流的 360 度视频预处理方法以及具体传输策略。

2.1 360 度视频的预处理

球形的原始 360 度视频在传输前往往需要被投射到二维平面上,由于用户在同一时间内只能关注整个平面上的一部分图像内容,因此 360 度视频的预处理主要是指合理将视频在空间上进行切分,这样在传输时才能区别对待用户可见的分区和暂时不可见的分区,从而达到减少传输数据量的目的。目前,主流的 360 度视频预处理方法可以被分为网格式的切分方法和基于视角的切分方法两类。

2.1.1 网格式的切分方法

许多工作都采用基于视野(Field of View, FOV)引导的切分方式,其中最基础的即为网格式(tiling)切分,它是指将 360 度视频在空间上分为多个平铺的网格片,在传输过程中仅对在用户视角内的网格片进行高质量传输,而剩余的平铺网格片则以较低质量传输或根本不传输。Mangiante 等人提出了一种移动网络边缘的视野(FOV)渲染解决方案^[8],旨在优化 VR 360° 视频流所需的带宽和延迟,其初步测试结果表明,这种方法在节省带宽方面的直接好处可以为 VR/AR 网络研究提供并产生新的方向。Gaddam 等人^[9]发现在实时交互式全景视频系统中一个主要的挑战在于这种用户虚拟视图的移动,即客户的兴趣区域会动态地、独立地变化,因此他们在平铺网格式切分方式的基础上为每一网格片都生成三种不同分辨率下的形式,如图 2.1 所示,这一方法能够降低带宽需求,并且提供与完整全景系统类似的体验质量。Ran 等人提出了一种新的基于超宽视角的方法^[10],以低带宽和低计算成本在互联网上传输高质量的 360 度视频,同样将源视频分割成小的网格,这些网格可以在用户端灵活地重建任意视图。其次,根据用户通常以低速交互的事实,他们扩展了用户要求满足实时交互需求

的观点, 还提供了一个低分辨率的全视图流, 以处理高速视图更改等异常情况。其结果表明, 在提供与本地虚拟现实相同的体验质量的同时平均节省了 60% 以上的带宽。

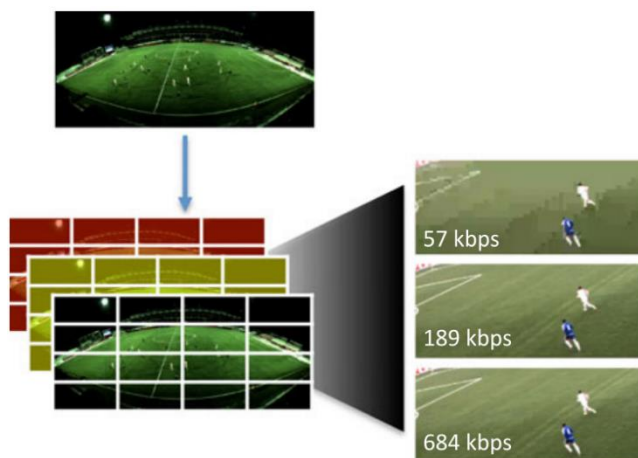


图 2.1 原始视频网格格式切片并生成不同分辨率的帧示例^[9]

2.1.2 基于视角的切分方法

除网格格式切分方式之外, 还有一些工作将 360 度视频编码为多个版本, 如图 2.2 所示, 这些不同版本具有不同的高质量区域, 客户端可以根据用户的当前观看方向接收适当的版本^[11], 这一基于了解用户视角的方法被称作视角版本式 (versioning) 切分。但是, 虽然这一方法可以减少 360 度视频流的带宽需求, 但它们不能减少延迟, 这是因为在确定用户视角后仍需要实时进行渲染和编码。

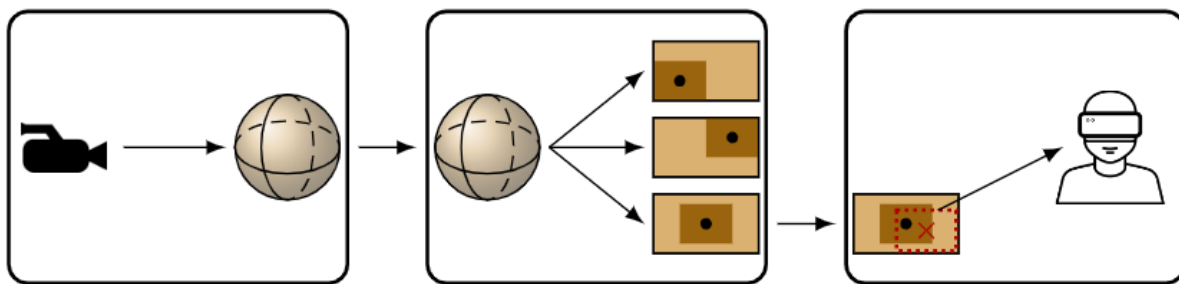


图 2.2 同一原始视频根据不同用户视角生成的多个版本^[11]

2.1.3 综合的切分方法

Sun 等人结合前文中所提到的网格格式切分方式和视角版本式切分方式, 提出了一个双层的 360 度视频流切分框架^[12], 如图 2.3 所示。首先将 360 度视频流划分为多个不重叠的时间段, 再将每个时间段内的视频图像数据编码为一个基本块 (Base Tier) 和多个增强块 (Enhancement Tier), 其区别在于基本块仅以低比特率对整个 360 度视图

范围进行编码以提供用户基本体验质量,而每个增强块在以特定方向为中心的特定视图覆盖视图窗口内对视频进行编码。文章指出将 360 度视频这样切分的好处是便于分配传输路径和进行缓冲,即在任一时刻,未来某一段时间段内的基本块可以被预取到一个长的流式缓冲区中以应对网络带宽的变化,这样可以保证在任何视角下用户都能以最低分辨率观看,不需要进行暂停或出现卡顿。在此基础上,每个增强块可以提供部分视角内高分辨率的图像以形成质量差异。同时,根据 Gaddam 等人的工作^[9],可以为同一个视图窗口生成多个以不同的比特率编码的增强块。另一个显著区别是同一时间段内的增强块视图窗口会相互重叠,覆盖整个 360 度视图范围,以实现完全覆盖和平滑过渡。

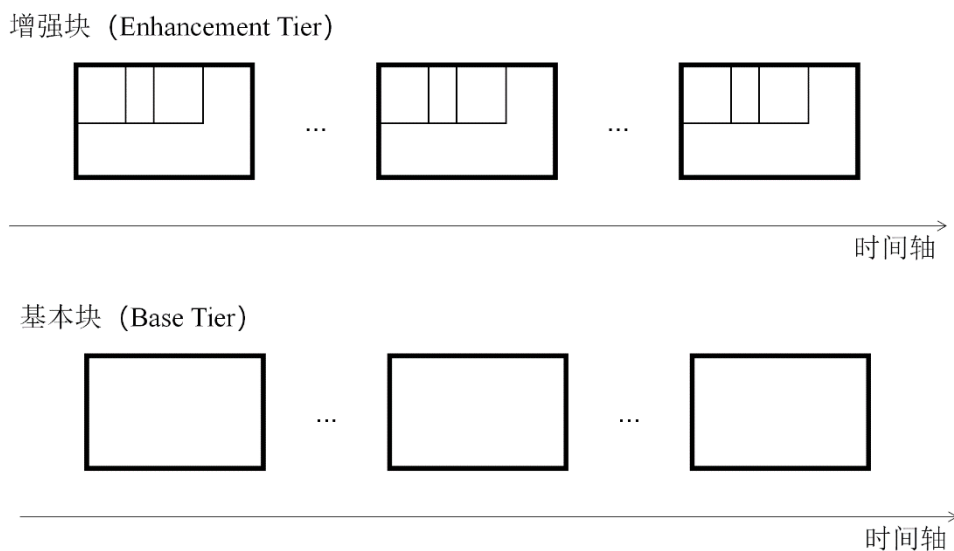


图 2.3 360 度视频的双层切分结构

2.2 360 度视频的传输机制

对 360 度视频进行预处理后,就可以有针对性地对部分图像进行传输,也可以采取新的传输策略。一些研究指出了 360 度视频和虚拟现实应用传输中的超低延迟和超高带宽挑战,Ge 等人提出了一种具有软件定义网络架构的多径协作路由方案^[13],以减少 5G 小蜂窝网络中 VR 无线传输的延迟和能耗。Liu 等人还研究了一种新的链路调度和自适应方案^[14],以减少 VR 视频流中的系统延迟和能耗。Hou 等人提出的自适应流媒体方法^[15]可以应用于任何现有无线网络而解决延迟和带宽挑战,而无需对前文提到的多径协作路由方案^[13]和链路调度自适应方案^[14]所要求的网络进行任何特殊的修改。

随着机器学习的逐渐流行和深度学习的不断发展,一些工作尝试通过预测用户视点

移动而获得更精确的高分辨率图像传输范围，在同等带宽下获得更好的用户体验。视点预测和相关的可信度预测都属于序列预测的问题，它被定义为预测给定历史序列的下一个值的问题^[16]。目前序列预测的方法可大致归纳为两类：一是传统的机器学习方法，二是深度学习方法。一些研究已经表明，传统的机器学习方法，如随机决策树^[17]和随机森林^[18]对许多序列预测任务都具有较为快速和有效的性能^[19]，Bootstrap 聚合决策树(BT)^[17]是其中最有效的方法之一。另一方面，近年来逐渐流行的深度学习方法，包括递归神经网络(RNN)及其变体和 LSTM 网络^[20]等，都已被证明在序列预测任务上表现良好^[21]。除此之外，还有一些研究使用深度神经网络，包括深度信念网络(DBN)^[22]和堆叠稀疏自动编码器(SAE)^[23]来实现序列预测。在这些深度学习方法中，LSTM 递归神经网络显示出捕捉人类运动过渡规律的良好潜力，如图 2.4 所示，这是因为它们具有学习观察训练数据之间时间依赖性的记忆^[24]。

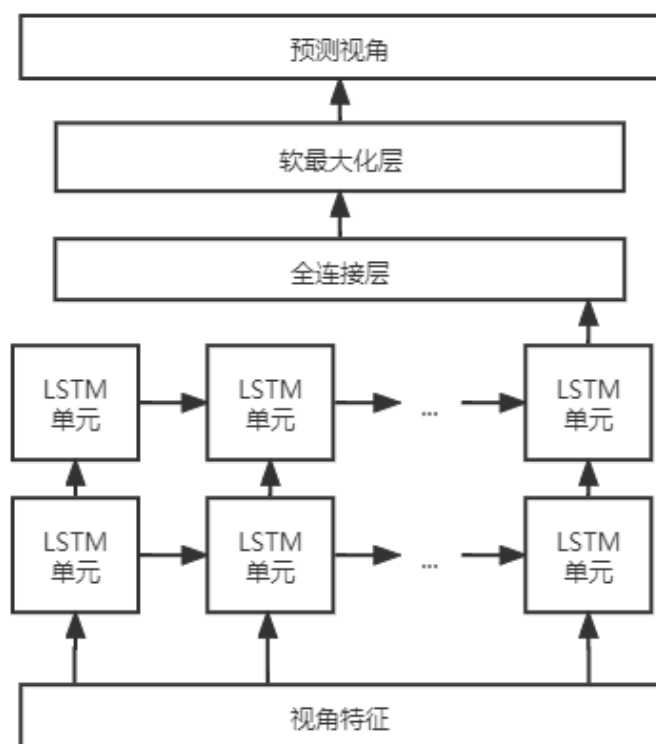


图 2.4 用于预测视点的 LSTM 网络结构

此外，一些研究探讨了头部运动预测的可行性。这些工作中的大多数都使用相对简单的模型，以欧拉角或角速度作为输入，而没有使用前文中所提到的切分方式及其对应的透视图。Qian 等人对商业 360 度视频平台进行了测量研究并提出了一种蜂窝友好的



流媒体方案^[25], 该方案基于头部运动预测, 只提供 360 度视频的可见部分, 真实用户收集的数据表明, 基于跟踪驱动的模式可以将带宽消耗减少多达 80%。Xu 等人调查了更复杂的预测模型^[26], 以有利于 360 度视频体验, 其使用由额外的眼动跟踪器收集的显著性图和凝视轨迹进行凝视预测。然而, 凝视预测技术无法直接实现, 因为大多数现有的 HMD 无法跟踪凝视, 并且其中的预测模型耗时较长, 因为从视频中提取图像显著性图和运动图需要更多的处理时间。

然而, 以上这些工作和研究往往都停留在理论分析和仿真实验的层面上, 缺乏在物理设备上的实现过程以及在实机场景中的真实测试。本文将所提出的预处理方法以及相应的传输机制在实机系统上进行了实现, 并可以作为对其他传输机制进行真实测试的基础平台。



3 多路径传输机制设计

本文主要通过将原始 360 度视频预先分块分段并使用可伸缩视频编码分层的方式进行预处理以形成静态资源,在此基础上通过同时使用多条网络链路带宽进行多路径传输的方式降低实时传输所需的网络带宽。本章将首先介绍传输机制的性能指标设计,并以此为出发点分别从 360 度视频的预处理与编码、360 度视频块的请求与应答过程以及 360 度视频块的路径调度策略三个主要方面详细介绍本文提出的具体方法。

3.1 传输机制的性能指标设计

本文的设计目标是通过将 360 度视频分块分层并进行多路径传输的方式最终提高用户体验质量。从网络的角度来看,在用户体验质量的度量上主要有两个评估指标,分别是视频播放的卡顿频率和供用户真实观看的有效流量。以下将依次介绍卡顿频率和有效流量的具体物理意义以及它们在实验中的具体算法。

3.1.1 卡顿频率

视频的卡顿(freeze)是真实用户在观看视频过程中较为不希望出现的情况之一,因而视频播放过程中卡顿的频率可以在一定程度上表征用户体验质量。然而,视频的卡顿可能由网络条件不佳、视频解码不及时、视频文件损坏等多种原因导致,为了避免其它因素的影响,尤其是所实现的实时系统中可伸缩视频编码的解码器耗时的影响,本文的实验中客户端将在每一时间间隔结束时立即,即解码和拼接等过程之前进行该时间间隔是否卡顿的判定,以此期望所统计的卡顿情况仅与网络条件相关。

具体而言,在每一时间间隔结束时,客户端将根据本时间间隔开始时向服务端请求的需要传输的视频块序列对照已成功完成传输的视频块层文件进行检查。若序列中存在某一基础层没有完成传输的视频块,则该时间段内不能以最低码率生成一个完全覆盖用户视野的视频段提供给用户进行观看,即认为该时间段 t 内发生卡顿,记 $p_t = 1$, 反之则该时间段内可以至少以最低码率生成一个完全覆盖用户视野的可观看的视频段,即该时间段内没有发生卡顿,记 $p_t = 0$ 。

综上所述,本文所使用的卡顿频率指标定义见式(3.1)。

$$\text{stall} = \sum_{t=1}^T p_t \quad (3.1)$$

其中, T 为视频总时长。

3.1.2 有效流量

在传统视频的实时传输系统中, 往往可以利用网络流量这一指标对用户观看到的视频质量进行评价, 这是因为在相同的编码条件下, 质量更好(分辨率、码率、帧率等更高)的视频一定需要更大的存储空间, 因而在实时系统的传输过程中也需要更高的网络带宽。然而, 对于 360 度视频而言, 由于用户在每一时刻都只能观看整个 360 度球形视频中的一部分内容, 因此只有用于传输用户真实观看到的视频内容的这一部分网络流量才是有效的, 可以作为对用户观看到的视频的质量评价。相对地, 用于传输用户没有观看的视频内容的这一部分网络流量对于用户而言是无效的, 这一部分无效网络流量不能作为用户观看到的视频的质量评价。

由于本文所实现的传输系统的服务端从整体逻辑功能上而言主要提供的是静态文件的下载服务, 该服务所产生的网络流量大部分都由所需传输的文件构成, 因此, 为了屏蔽网络连接建立和释放等过程中产生网络流量的影响, 本文直接使用客户端所收到文件的大小代替网络流量进行统计和分析。在此基础上, 前文中对于网络流量有效和无效的界定同样适用于客户端所收到的视频块层文件, 即仅将与用户真实视野有重叠区域的视频块的各层文件视为有效, 将其大小加权后参与用户观看到的视频的质量评价。

综上所述, 本文所使用的有效流量指标定义见式(3.2)。

$$\text{traffic} = \sum_{t=1}^T \sum_{i=1}^N \left(w_{i,t} \sum_{l=1}^L s_{t,i,l} \right) \quad (3.2)$$

其中, T 为视频总时长, N 为原始 360 度视频每一视频段在空间上被切分出的视频块数量, L 为每一视频块经过可伸缩视频编码的层数, $s_{t,i,l}$ 为客户端收到的第 t 个时间段的第 i 个视频块的第 l 层的文件大小, $w_{i,t}$ 为第 i 个视频块在第 t 个时间段内的权值, 这一权值的物理意义是该视频块与当前时间段内用户视野重叠部分的面积占该视频块面积的比例。由于这一权值的求法较为重要, 因此在这里给出求两个矩形相重叠区域面积的具体算法如算法 1 所示。

算法 1 求两个矩形重叠区域的面积

输入： 矩形一对角线端点坐标 $(x1, y1), (x2, y2)$, 矩形二对角线端点坐标 $(x3, y3), (x4, y4)$

输出： 两个矩形重叠区域的面积 S

```

1.  $x5 \leftarrow \max(\min(x1, x2), \min(x3, x4))$ 
2.  $y5 \leftarrow \max(\min(y1, y2), \min(y3, y4))$ 
3.  $x6 \leftarrow \min(\max(x1, x2), \max(x3, x4))$ 
4.  $y6 \leftarrow \min(\max(y1, y2), \max(y3, y4))$ 
5. if  $x5 < x6$  and  $y5 < y6$  then
6.    $S \leftarrow (x6 - x5) * (y6 - y5)$ 
7. else do
8.    $S \leftarrow 0$ 
9. end if
10.
11. return  $S$ 

```

3.2 360 度视频的预处理与编码

这一部分的主要目的是将 360 度视频以利于请求的形式作为静态资源存放于服务器上。在这一部分中，首先需要将所下载的 360 度视频转换成合适的格式，然后在此基础上对 360 度视频进行分块等预处理，最后再使用可伸缩视频编码（SVC 编码）将其分层后作为静态资源存放于服务器上。经过这一系列流程，最终完成原始 360 度视频到服务器上易于请求的静态资源形式的转变。其整体处理流程如图 3.1 所示。

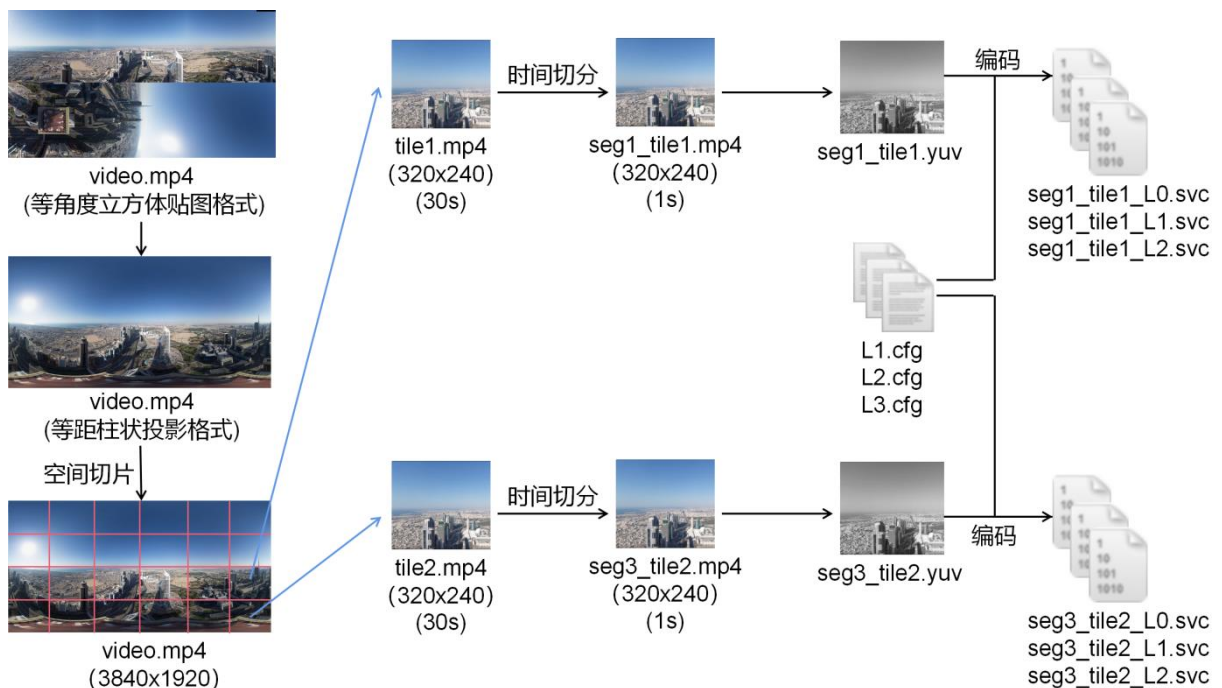


图 3.1 360 度视频的预处理与编码过程

3.2.1 360 度视频的格式转换

360 度视频是一种球形视频, 在进行进一步的处理和传输之前, 一般需要将其投影到 2D 平面上, 不同的投影方式也即不同的 360 度视频格式。目前主流的 360 度视频格式有以下三种: 等距柱状投影格式 (Equirectangular)、立方体贴图格式 (Cubemap) 等角度立方体贴图格式 (Equi-Angular Cubemap)。

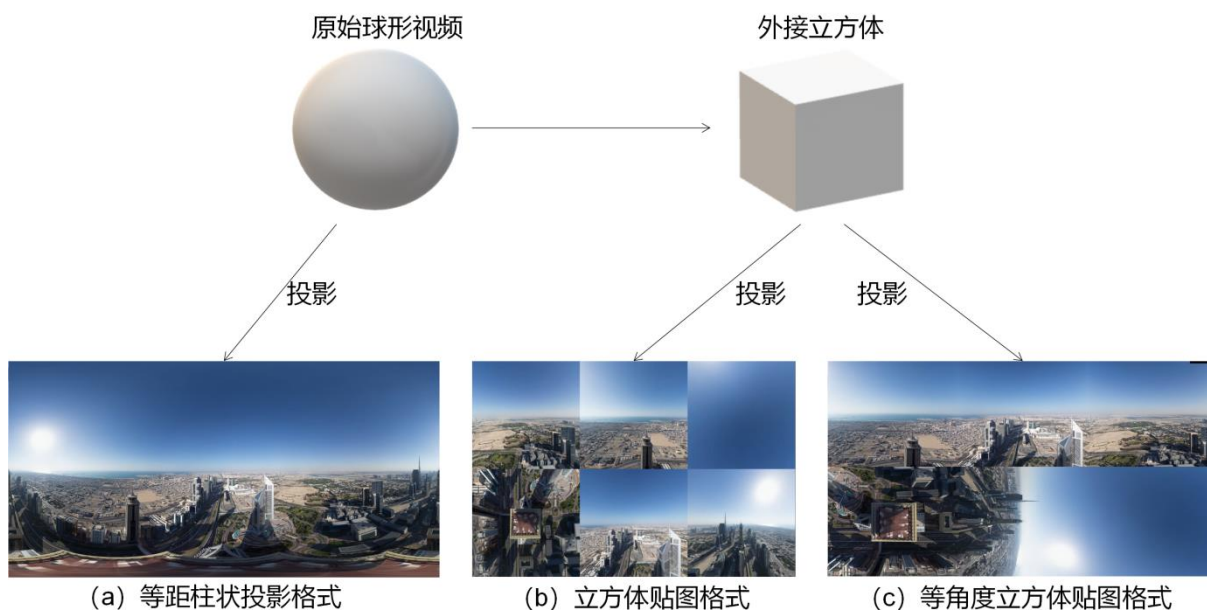


图 3.2 360 度视频的三种常见格式

等距柱状投影格式与世界地图的基于经纬度的投影方式类似, 它的做法是将球体的各条经线和各条纬线等距或按照不同疏密程度投影到 2D 平面上, 如图 3.2 (a) 所示。由于等距柱状投影格式在投影形成的 2D 平面内部没有图像的突变, 因此是最直观的投影方式。但按照这种方式投影的视频在南北两极投影出的像素数量过多而赤道附近投影出的像素数量较少, 因此, 等距柱状投影格式的主要缺点即是其 2D 图像重建为球形视频时清晰度损失较大。

立方体贴图格式也是一种常见的 360 度视频格式, 与直接从球形投影到 2D 平面不同, 它的做法是首先将球体上的内容投影到其外接立方体上, 然后将这一立方体展开为六个面, 并平铺在 2D 平面上, 如图 3.2 (b) 所示。这一格式相对于等距柱状投影格式而言图像的扭曲更小, 但是其所生成的立方体各个面的边角处距离原始球体较远, 因此所对应的图像像素分布比仍然不太均匀。

等角度立方体贴图格式是谷歌所提出的一种较新的格式, 与前文所提到的传统立方

体贴图格式相比,它的优化点在于更改向外接立方体投影时的采样点位置,如图 3.2(c) 所示,这一做法使得立方体各个面的边角与中心所对应图像的像素分布更加均匀,因此在重建为球形视频时可以提高一些细节的清晰程度。

在本工作中,由于需要对 360 度视频进行较细粒度的分块传输,因此我们选取在投影形成的 2D 平面内部没有图像突变的等距柱状投影格式作为后续切片等处理的输入格式。而从 Youtube 等在线视频网站上直接下载的 360 度视频目前普遍为等角度立方体贴图格式,因此我们需要首先进行 360 度视频进行格式的转换,从等角度立方体贴图格式转换为所需的等距柱状投影格式。

3.2.2 360 度视频的切分

360 度视频是一种球形视频,以等距柱状投影的方式生成的 2D 图像包含整个原始球形视频各个方向上的信息。而由于用户在每一时刻都只能观看到视频的一部分而非全部,因此对于每一时刻而言,在传输过程中都不需要传输该时刻所对应视频帧的全部图像,只需要传输全景中的一部分图像即可。为了实现这一需求,一个主要的思路即是对等距柱状投影格式的 360 度视频进行空间上的切分,即将每一帧图像均预先切分为若干视频块(tile),这一做法使得传输过程中服务端只需要发送用户当前视点周围的一系列视频块,从而与传输整个视频画面的做法相比占用更少的网络带宽。

视频块的大小对于传输系统的效果有很大影响。如果视频块过小,原始视频所切分出的视频块数量过多,会造成客户端拼接视频块过程耗时的增加,从而可能在应用层面造成播放的卡顿。此外,由于服务端仅能根据当前时刻及之前的用户视点预测下一时刻用户视点并决定发送的视频块序列,因此当这一预测方法与实际情况有一定误差时容易引起所发送视频块不足以覆盖真实用户视野的问题,从而造成用户体验的下降。如果视频块过大,原始视频所切分出的视频块数量过少,则能够完全覆盖所预测用户视野的视频块序列整体占原始视频的比例过大,使得这一仅传输部分视频块的方法不具有相较于传输整个视频画面做法的明显优势。

目前 360 度视频往往以较高的分辨率进行存储,通常为 4K, 8K, 甚至 16K。为了实验的方便,也由于目前在线视频网站下载器的限制,本文选用分辨率为 4K(3840*1920) 的 360 度视频作为原始视频进行预处理和传输。为了避免前文中所提到的视频块过大或



过小造成的问题,本文以 320×240 分辨率作为每一视频块的分辨率,在此条件下,整个等距柱状投影格式的原始 360 度视频在 2D 平面上横向被分为 12 份,纵向被分为 8 份,即共 96 个视频块。

由于用户在不同的时间点上观看的视频位置(即用户视点)不同,因此对于已在空间上进行切分的 360 度视频也需要进行时间上的切分,即切分成若干视频段(segment)。与视频块大小的选取类似,视频段的长度也对传输系统的实际效果有较大影响。如果视频段长度太大,则用户视点的更新不能及时反映到服务端对视频块的选取这一过程中,造成所传输视频块序列与用户真实视野之间的较大差异,可能会出现卡顿、黑屏等情况从而降低用户体验。本文以 1s 作为单位时间进行原始视频时间上的切分过程,即每一个视频段长度均为 1s。

经过这一系列处理,完成了原始 360 度视频在时间和空间上的解构,生成的视频片段将作为后续工作和实验的基本单位。

3.2.3 360 度视频的编码

360 度视频的特点是分辨率和码率均较高,虽然已经将原始视频在空间上分为多个视频块,但如果使用传统的视频编解码技术,传输的视频块文件仍然较大,将会占用较多网络带宽,难以达到目的,因此 360 度视频需要一种特殊的编解码方式。

H.264 标准的其中一个拓展——可伸缩视频编码(Scalable Video Coding, SVC)技术是一种用来兼容不同终端设备和链路带宽的编码技术,这一技术可以利用原始的传统视频生成一个具有多个不同质量层级的可分层视频流,这些层之间的分辨率、码率、帧率等质量指标有所区别。在生成的这些层中,质量最低的为基础层(Base Layer),其他层按照质量从低到高依次为各个增强层(Enhancement Layer)。总体而言,这种分层的编码方式对提升传输的误码弹性和视频质量等有正面意义。

可伸缩视频编码方式的特点是其解码过程中基础层可以单独解码,而各个增强层可以依赖基础层和更低的增强层进行解码。这一特点使得 SVC 编码具有一次编码、多次解码的优势,即解码过程可以根据实际层数进行。具体而言,当网络带宽不好时,服务端可以仅向客户端传输基础层以保证基本播放流畅度,随着网络带宽的提高,服务端可以逐渐提高传输的增强层数量,从而在客户端解码出更高质量的视频以提高用户体验质

量。因此，可伸缩视频编解码方式很适合 360 度视频传输的场景。

可伸缩视频编码支持三种不同的分级方法：时域分级、空间分级和质量分级。时域分级是指将视频各帧分为不同的等级，这样编码形成的各个层的帧率有所区别，一个示例如图 3.3 所示，其中 L0, L1 等分别表示各帧所在的不同层，解码时需要确定一个阈值，仅解码层数低于此阈值的视频帧。如图 3.4 所示，空间分级是指将视频编码为不同分辨率的多个层。而质量分级是指将视频按照不同的量化步长进行分层，其中低层的帧量化步长较长，而高层的帧采用更小的量化步长，这样高层帧可以对低层帧的错误进行纠正。

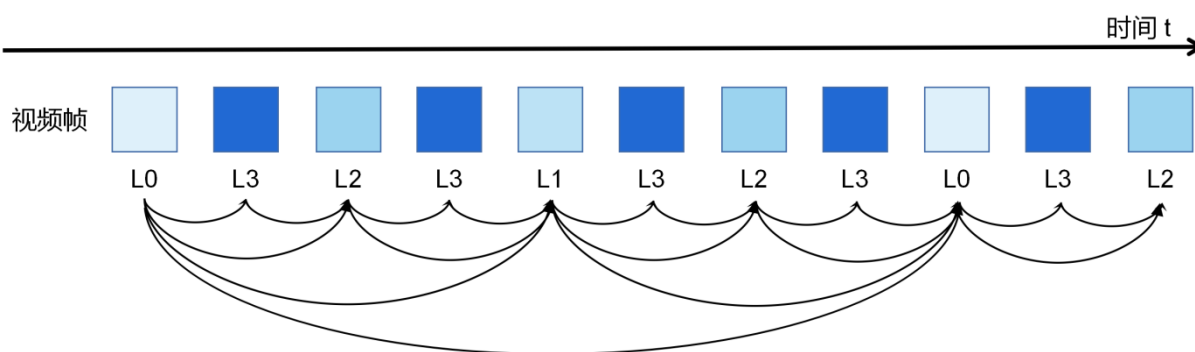


图 3.3 可伸缩视频编码的时域分级方式

由于质量分级方式具有抗差错性强的特点，本文将使用质量分级的方式对所选视频进行编码。为了实验的简便以及不失一般性，本文将已经在时间和空间上切分的 360 度视频分为一个基础层 L1 和两个增强层 L2、L3，其中基础层 L1 的量化步长为 40，增强层 L2 的量化步长为 30，增强层 L3 的量化步长为 20。

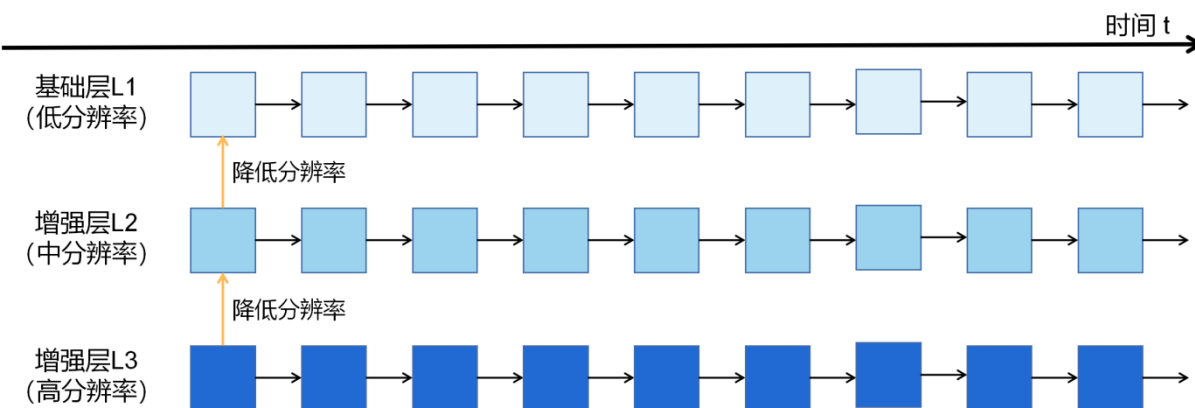


图 3.4 可伸缩视频编码的空间分级方式

3.3 360 度视频块的请求与应答过程

在介绍 360 度视频块的具体请求与应答过程之前, 首先给出本文的场景。如图 3.5 所示, 本文主要针对用户观看在线视频网站上的 360 度视频这一实际场景进行设计。在这一场景中, 360 度视频以文件形式作为静态资源存放于服务器上, 用户在客户端实时向服务端发送请求, 服务端根据请求的内容进行应答, 以此完成视频的传输过程。

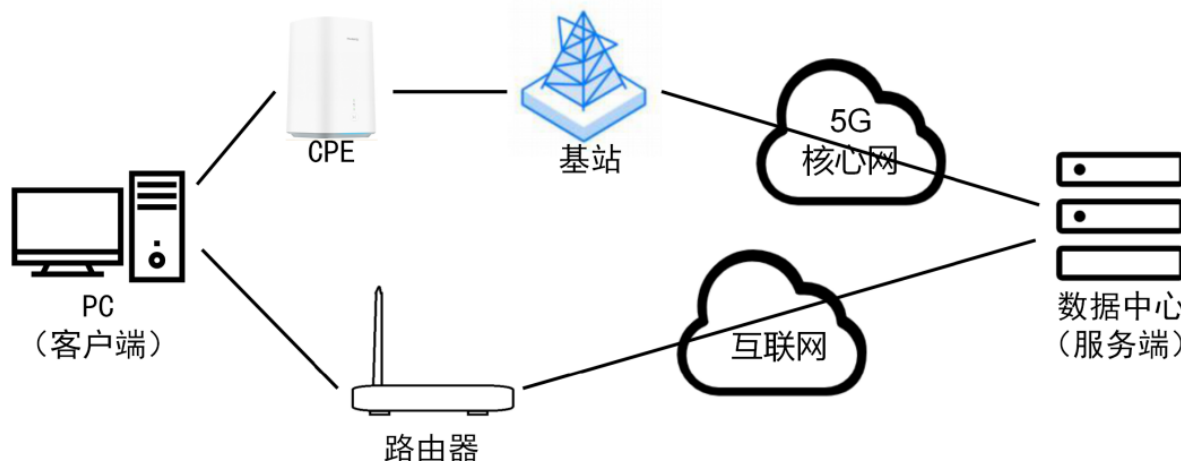


图 3.5 多路径传输场景

本文所实现的 360 度视频块的请求与应答过程如图 3.6 所示。系统整体上是一个实时系统, 以 1s 作为单位时间, 这与前文中 360 度视频在时间上切分出的视频段长度保持一致。以下将对该请求与应答过程进行具体阐述。

在每一单位时间开始的时刻, 客户端都会将当前的用户所观看的视点 (即用户视野的中心) 位置包含在请求包中发送给服务端。服务端收到这一请求包后解析出当前用户视点在原始 360 度视频中的横、纵像素坐标, 并以用户视点为中心确定一个分辨率为 $1280*720$ 的区域。在此基础上, 服务端遍历处于这一时间段内的所有视频块并挑选出与所生成区域相交的视频块, 以列表的形式将这些视频块的横、纵块坐标返回给客户端。在这一单位时间的剩余时间里, 客户端将依次请求处于列表中的这些视频块的基础层和各个增强层, 其中优先请求基础层以保障视频流畅度, 再逐层请求增强层, 而服务端根据客户端发送的这些视频块坐标和层数进行应答, 返回其对应视频块的对应层的数据, 客户端将这些数据进行存储。这一单位时间结束时, 客户端会检查所有已存储的视频块层文件, 将各个视频块均以传输完成的最高层级进行解码, 在此基础上按照各块的横、纵块坐标在空间上拼接在一起, 并以用户当前视点为中心, 以 $1280*720$ 分辨率进行裁

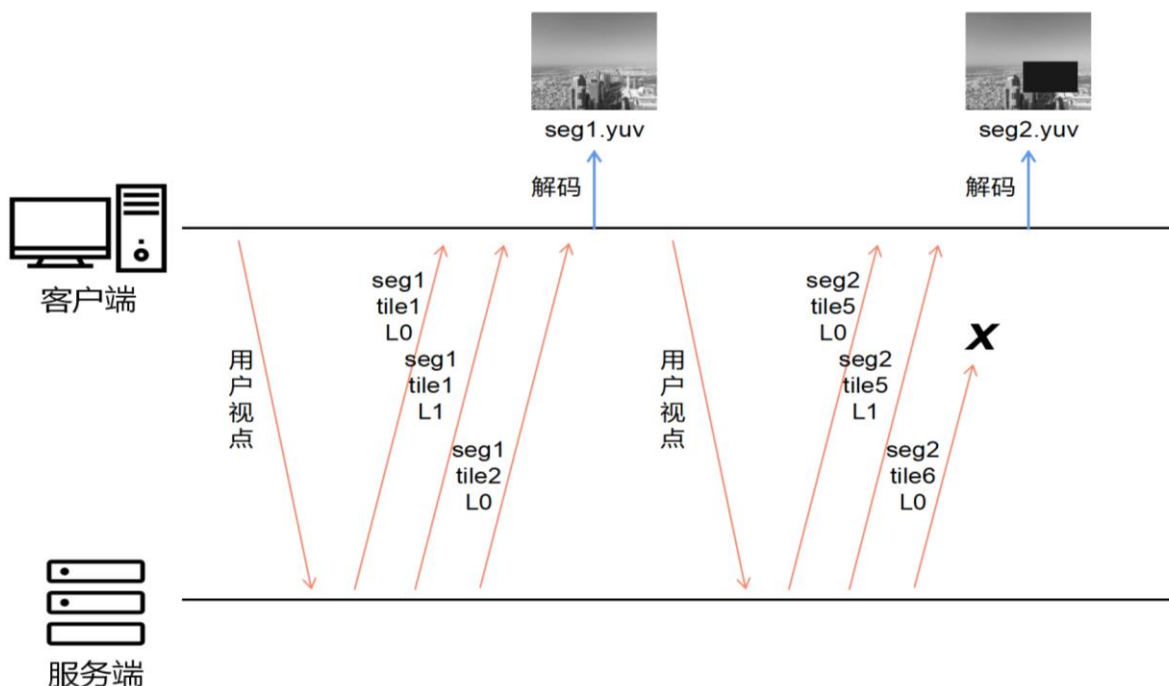


图 3.6 360 度视频块的请求与应答过程

剪，形成这一单位时间内传输完成的视频片段作为用户在其视野内所观看的视频。如果在这一单位时间结束的时刻仍有部分数据在链路上传输，则客户端将会在这些数据在之后的时间到达后进行丢弃。传输过程结束后，客户端将各个时间间隔内的视频片段进行拼接，形成整个视频并向用户播放。

3.4 360 度视频块的路径调度策略

在本文的工作中，可伸缩视频编码将预先分块分段的 360 度视频分为不同质量的一个基础层和两个增强层，这些层以文件的形式存放，并作为传输调度等过程的基本处理单位。

经可伸缩视频编码的视频具有基础层可以单独解码，而更高层只能依赖更低的所有层进行解码的特点，因此若某一时间段内基础层没有按时到达客户端，即使客户端已经成功收到该视频块的若干增强层，由于增强层不能在缺少基础层的情况下单独解码，用户在观看视频的这一刻只能出现卡顿（freeze）。这说明在前文所提出的两个主要评估指标中，传输基础层主要对降低卡顿频率有贡献，能够保障用户观看视频的基本流畅度。相比之下，增强层的文件大小远大于基础层的文件大小，在有效流量这一指标的计算中占比更大，因此传输增强层主要对提高有效流量有贡献，能够提高用户观看视频的质量。



本文提出了一种基础的 360 度视频块路径调度策略。由于已切分的 360 度视频块经可伸缩视频编码后生成的基础层的大小较小，传统的 Wi-Fi 链路已经可以完全支持基础层的实时传输，因此本文将使用 Wi-Fi 链路进行 360 度视频基础层的传输。这一做法的另一个好处在于 Wi-Fi 链路较为稳定，使用 Wi-Fi 传输基础层能够更好保障视频的基本流畅度，减少卡顿的发生，从而保障用户基本体验质量。另一方面，可伸缩视频编码生成的增强层大小较大，Wi-Fi 链路难以支持其实时传输，因此本文将使用 5G 链路进行 360 度视频增强层的传输，这样的调度策略可以在网络带宽条件较好时显著提升视频质量，从而显著提升用户体验质量。



4 基于 Wi-Fi 与 5G 链路的多路径系统实现

基于第三章介绍的适用于 360 度视频的多路径传输系统的设计中 360 度视频的预处理与编码、360 度视频块请求与应答过程以及 360 度视频块的路径调度策略等内容,本章将对相关内容进行实现。本章将首先对 360 度视频多路径传输系统的需求进行分析,并在此基础上提出本文所实现系统的整体架构,随后将依次对整体架构中的各个层次内的各个重要模块进行详细介绍。

4.1 系统需求分析

本文的主要目标是实现一个适用于 360 度视频的多路径传输系统,该系统主要通过将 360 度视频分块分层并进行多路径传输的方式减少实时传输 360 度视频所需要的网络带宽,从而降低视频卡顿频率,提高用户视口内的视频质量,进而提高用户体验质量。这一系统的功能需求如下:

1. 360 度视频静态资源生成。这一功能要求系统可以将从在线视频网站上下载的原 始 360 度视频处理为符合系统内部数据规范的易于请求的文件形式。其具体要求包括能够对原始 360 度视频进行时间和空间上的切分以生成粒度更小的视频片段,以及能够使用可伸缩视频编码(SVC)将已经完成切分的各视频片段进行分层。这一功能最终能够将预处理和编码过程生成的视频块层文件以静态资源形式存放于服务器上。

2. 用户视野判定与视频块选取。这一功能要求系统可以根据用户当前视点预测用户下一时间段内的视野,并以此为根据选取应当进行传输的视频块序列。本系统要求这一选取过程应当具有一定的鲁棒性,即当用户真实视点有所偏移时也能在一定程度上保证用户观看视频的基本流畅性。

3. 视频块层二进制文件的传输。这一功能要求系统可以将已编码的 360 度视频各块层的二进制文件从服务端传输到客户端。在此基础上,本系统要求需要对传输过程所用时间的进行控制,即仅将在一定时间间隔内完成传输的视频块层视为有效,而忽略在传输期限到来时仍未完成传输的各文件。

4. 解码与生成用户可观看的视频。这一功能要求系统可以根据已完成传输的各视频块层文件进行 SVC 的解码过程,并生成相对应的视频片段,在此基础上系统需要实现

各视频片段在时间和空间上的拼接与合并，最终形成用户可观看的视频。

4.2 系统总体架构

本文所设计的适用于 360 度视频的多路径传输系统可被分为物理链路层和应用业务层两个主要层次，如图 4.1 所示，下面将具体介绍各层次的主要模块与功能。

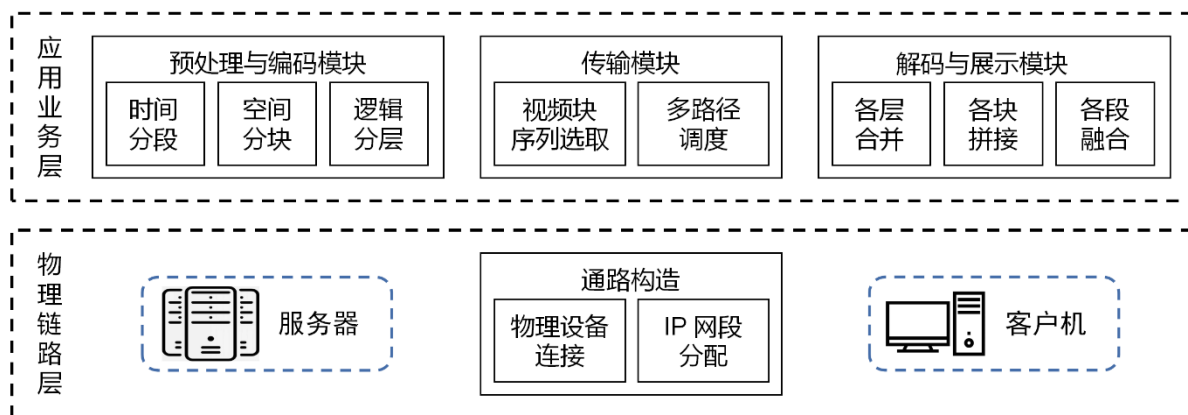


图 4.1 系统总体架构图

物理链路层主要指硬件层面上各物理设备的连接与调试，其中重点在于构造两条可用于传输的物理路径，并将这两条路径在逻辑上进行合并，完成多路径传输系统物理链路的搭建，并在此基础上对各个物理设备的各个物理网络接口进行 IP 网段的分配，实现全网的互联互通，最终实现普通数据流的多路径传输。物理链路层将作为应用业务层各模块的实现基础，提供物理层面的可行性。

应用业务层主要指软件层面上使得 360 度视频能够以所规定的形式进行多路径传输的各个主要模块。与物理链路层中服务器端相对应的是预处理与编码模块，这一模块可以将在线视频网站下载的原始 360 度视频进行时间上的分段、空间上的分块，并使用可伸缩视频编码 (SVC) 将其进行逻辑上的分层。与物理链路层中客户端相对应的则是解码与展示模块，这一模块可以将传输完成的各视频块层二进制文件在逻辑上进行层的合并、在空间上进行块的拼接，以及在时间上进行段的融合，最终生成可供用户观看的视频。将物理链路层上服务器端与客户机端相联系起来的即是传输模块，这一模块是本文所实现的适用于 360 度视频的多路径传输系统的重要模块，它主要实现了将以静态资源形式存放在服务器上的 360 度视频传输到客户机的过程，实现了服务器与客户机之间的请求与应答。该模块将实现数个重要算法以完成 360 度视频的多路径传输过程，

这其中包括根据用户视点选取需要传输的视频块索引序列的算法以及具体视频块在各路径上的调度策略。

4.3 物理链路层搭建与调试

多路径传输方式具有可同时使用多条物理链路带宽的优势,从而在一些大流量业务场景中具备一定优势。与传统的点对点单路径传输方法相比,多路径方法的主要难点在于同时连接多条路径的终端收到的包具有乱序的特点,需要进行一定的处理,而这一特点在系统中仅存在两条路径时已完全得以显现。因此,不失一般性地,本文将搭建一个具有两条物理链路的多路径传输系统。

第五代移动通信技术(5G)是一种具有高速率、低时延和大连接特点的新一代宽带移动通信技术,在近几年吸引了很多关注。目前,它已经被初步应用于一些 4G 难以满足要求的大流量业务场景,这其中也包括 360 度视频的传输场景。因此,本文将 5G 链路作为多路径系统的其中一条路径。但是,目前 5G 技术仍处于起步和发展阶段,因而其定价较高,长时间大流量使用 5G 网络将会很昂贵。相比较而言,传统无线局域网 Wi-Fi 技术成本较为低廉,且更加稳定,是目前主流使用的网络技术之一。因此,本文将 Wi-Fi 链路作为多路径系统中的另一条路径。

在确定物理设备并完成连线后,本文按照图 4.2 完成了 IP 网段划分,实现了多路径传输系统的物理结构。其中,客户端为一台 PC 机,IP 地址为 192.168.8.100,子网掩码为 255.255.255.0;服务端位于数据中心服务器的一台虚拟机上,该虚拟机映射到两张物理网卡,IP 地址分别为 192.168.242.200 和 192.168.246.1,子网掩码均为 255.255.255.0。Wi-Fi 逻辑链路的实现上,PC 与一台路由器通过无线连接,路由器对

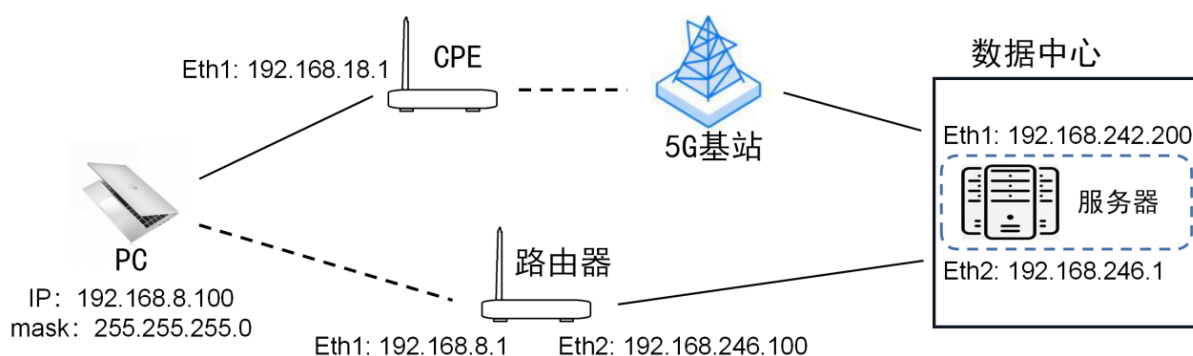


图 4.2 多路径传输系统的网段划分



应接口的 IP 地址为 192.168.8.1，该路由器的另一接口（IP 地址为 192.168.246.100）与数据中心通过双绞线连接，对应服务器物理网卡的 IP 地址为 192.168.246.1。5G 链路的实现上，PC 与用户前置设备 CPE 通过双绞线连接，CPE 对应接口的 IP 地址为 192.168.18.1，CPE 的另一接口与 5G 基站通过 5G 信号相连接，5G 基站另一方向上与数据中心通过有线方式连接，其对应的服务器物理网卡 IP 地址为 192.168.242.200。

完成这一多路径传输系统的物理链路搭建并为各个设备的各个网络接口设置对应的 IP 地址以及子网掩码后，还需要在适当的位置添加合适的静态路由以实现客户端和服务端的互联互通。在本文所搭建的物理链路中，PC 和 CPE 直接连接的两个接口 IP 地址不在同一网段，不能实现通信。因此，本文在 PC 上添加如表 4.1 所示的两条永久静态路由，最终实现了多路径传输系统的互联互通。

表 4.1 PC 上添加的静态路由

网络地址	子网掩码	下一跳
192.168.246.0	255.255.255.0	192.168.8.1
192.168.242.0	255.255.255.0	192.168.18.1

4.4 应用业务层关键模块实现

4.4.1 360 度视频的预处理与编码

根据前文中对 360 度视频静态资源生成过程的设计，本文将分别通过四个步骤进行实现：格式转换、空间分块、时间分段、SVC 编码。这一模块的代码实现如算法 2 所示。

多媒体视频处理工具 FFmpeg 是一个功能强大、用途广泛的开源视频处理库。它不仅支持普通视频的转换、合并以及分解等操作，可用于实现 360 度视频空间上的分块和时间上的分段过程，并且 FFmpeg 在 4.3 版本中加入了一个适用于 360 度视频的过滤器，可用于实现 360 度视频的格式转换。因此，这一模块的前三个步骤，即格式转换、空间分块、时间分段均通过 FFmpeg 的命令行工具实现。

可伸缩视频编码 SVC 目前的应用较少，缺乏广泛流行和使用的编解码器。因此本文将在 SVC 编码的参考实现代码 JSVM 的基础上对 360 度视频进行 SVC 编码。具体而言，JSVM 提供了 H264AVCDecoderLibStatic 等编解码库和相关工具，本文主要需要参考编解码器所给出的示例配置文件编写自己的配置文件，再使用对应的编码库将已在空间上分块和在时间上分段的 360 度视频编码前文所设计的一个基础层和两个增强层。经过编码

后的视频各层均单独以文件形式作为静态资源存放于服务器上,文件名中将包含其在原始 360 度视频中的时间段、空间块横、纵索引以及层数信息。

算法 2 360 度视频静态资源的生成算法

输入: 原始 360 度视频 V , 时长为 T , 分辨率为 3840×1920

输出: SVC 层序列 $A[T][N][M][L]$

```
1.   $V \leftarrow \text{ffmpeg.convert\_EAC\_to\_E}(V)$     // 使用 ffmpeg 将原始视频转为等距柱状投影格式
2.  for  $i = 0$  to  $T - 1$  do
3.       $\text{seg}[i] \leftarrow \text{ffmpeg.time\_split}(V, i, 1)$     // 使用 ffmpeg 将原始视频以 1s 为单位时间切割
4.  end for
5.  for  $i = 0$  to  $T - 1$  do
6.      for  $x = 0$  to  $N$  do
7.          for  $y = 0$  to  $M$  do
8.               $\text{tile}[i][x][y] \leftarrow \text{ffmpeg.split}(\text{seg}[i], x \times 320, y \times 240, 320, 240)$     // 使用 ffmpeg 将
              每一视频片段按  $320 \times 240$  为单位分辨率进行空间上切割
9.          end for
10.     end for
11. end for
12. for  $i = 0$  to  $T - 1$  do
13.     for  $x = 0$  to  $N$  do
14.         for  $y = 0$  to  $M$  do
15.              $\text{config}[] \leftarrow \text{gen\_cfg}(i, x, y, L)$     // 生成基础层和增强层对应的配置文件
16.              $A[i][x][y][L] \leftarrow \text{JSVM.encode}(\text{tile}[i][x][y], \text{config}[])$     // 编码
17.         end for
18.     end for
19. end for
20.
21. return  $A[T][N][M][L]$ 
```

4.4.2 服务端

根据上一章中的 360 度视频多路径传输机制设计与上一节所述的系统整体架构,服务端共需实现两个重要接口,分别为视频块层文件下载接口和视频块序列选取接口。其中,视频块层文件下载接口的逻辑较为简单,只需要根据客户端请求包中的时间、空间索引返回对应的视频块层文件即可,而视频块序列选取接口则需要实现一个相对复杂的序列选取算法,这一算法如算法 3 所示。

算法 3 视频块序列的选取算法

输入：用户视点横、纵坐标 (x, y)

输出：视频块选取矩阵 $A[N][M]$ ，其中 $A[i][j]$ 若为 **true** 则视频块 (i, j) 被选取

```

1. flow  $\leftarrow$  check_flow( $x, y, 1280, 720$ )    // 检查用户视野区域是否超出横坐标边界
2. (area_left, area_right, area_up, area_down)  $\leftarrow$  calculate_boundary( $x, y, 1280, 720$ )    // 计算用户视野区域顶点坐标
3. for i = 0 to N do
4.     for j = 0 to M do    // 遍历视频块，判断是否与用户视野区域相交
5.         A[i][j]  $\leftarrow$  true
6.         if j * 240 > area_down or j * 240 + 240 < area_up then
7.             A[i][j]  $\leftarrow$  false
8.         end if
9.         if not flow then
10.            if i * 320 > area_right or i * 320 + 320 < area_left then
11.                A[i][j]  $\leftarrow$  false
12.            end if
13.        else do
14.            if i * 320 > area_right and i * 320 + 320 < area_left then
15.                A[i][j]  $\leftarrow$  false
16.            end if
17.        end if
18.    end for
19. end for
20.
21. return A[N][M]

```

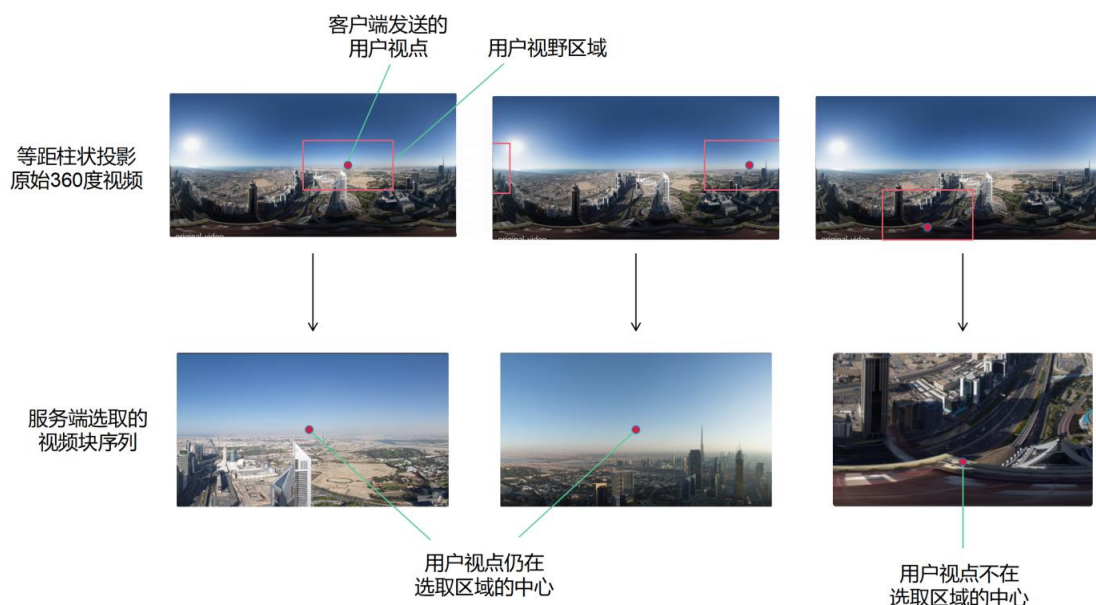


图 4.3 不同用户视点对应的视频块序列示例

值得注意的是,如图 4.3 所示,由于等距柱状投影格式的 360 度视频在横向边界上具备良好的可拼接性,即这一格式不改变右边界上的视频块和左边界上的视频块在原始球形 360 度视频中的相对位置,因此若用户视点处于横向边界附近,服务端将通过越过边界从另一方向上进行补足的方式保证所预测的用户视野区域以客户端发送的用户视点为中心。与之相对的,等距柱状投影格式的 360 度视频在纵向边界上不具有这一特性,其纵向边界上的视频块分别对应原始球形 360 度视频南、北两级的内容,因此若用户视点处于纵向边界附近,服务端将显式保证所预测的用户视野区域不纵向越过边界,也即所预测的用户视野区域不再以客户端发送的用户视点为中心。

4.4.3 客户端

根据上一章中 360 度视频多路径传输机制设计与上一节的系统整体架构,客户端需要实现从获取用户视点到利用客户端返回的视频块层生成用户可观看的视频这一过程。

算法 4 客户端的请求与视频生成过程

输入: 用户视点序列 $P[T]$, 其中 T 为总时长

输出: 用户可观看的视频 V

```
1.  for seg = 0 to T do
2.      t ← 0
3.      tile_list[] ← request_tile_list(P[seg])    // 向服务端请求视频块列表
4.      while t < 1 do    // 仅将在 1s 内完成传输的内容视为有效
5.          download[i][j][l] ← request_next_tile(tile_list[])
6.          t ← t + consume_t
7.      end while
8.      for i = 0 to N do
9.          for j = 0 to M do
10.             tile[i][j] ← svc_merge(download[i][j])
11.             tile[i][j] ← decode(tile[i][j])    // 将传输得到的二进制文件解码为视频
12.          end for
13.      end for
14.      video[seg] ← tile_merge(tile)    // 将位于同一时间段的视频块进行拼接
15.      video[seg] ← crop(video[seg], P[seg])    // 将视频根据用户视点进行裁剪
16.  end for
17.  V ← Time_merge(video[])    // 将各时间段内的视频进行拼接, 形成最终用户可观看的视频
18.  return V
```



具体而言,对于每一时间段,客户端所实现的过程可主要分为四步:请求需要传输的视频块索引列表、依次请求列表中各视频块的各层数据、将同一视频块各层数据合并并解码形成视频、将列表内视频进行空间上的拼接与裁剪。具体实现上,与前文 360 度视频静态资源的生成过程类似的,客户端也使用 FFmpeg 这一视频处理库将解码完成的视频进行空间和时间上的拼接操作,以及将视频根据用户视点进行裁剪的过程。解码器的选择上,本文参考 DASH-SVC-Toolchain 的方法使用 JSVM 的相应库进行解码,此解码过程分为将各层进行合并以及将二进制文件解码为视频这两个主要步骤。客户端的相关实现如算法 4 所示。

5 系统测试与实验结果

基于第三、四章介绍的适用于 360 度视频的多路径传输系统的设计与具体实现，本章将对相关内容进行具体实验。本章将首先介绍实验所用的测试数据，并展示系统所实现功能的具体效果，最后将在前文所介绍的两个重要评估指标的基础上对性能测试结果进行分析，从而给出本文提出的方法相较传统方法的优势所在。

5.1 测试数据

研究指出^[27]，在线视频网站上的 360 度视频主要可以被分为 5 类：探索、静态焦点、移动焦点、快速前进以及杂项。探索类 360 度视频往往由类似于位于俯瞰城市的高层建筑顶部的摄像头等进行拍摄。用户在观看这类 360 度视频的过程中往往没有特别感兴趣的特定对象或方向，而是希望在整个视频的时间内持续探索整个 360 度领域，因此用户视点的移动较为频繁和广泛，很适合用于传输策略的评价。此外，探索类 360 度视频的另一个特点是两个独立观众在观看视频的每个时间点上所预期观看的视角在实质上不同，因此很难使用机器学习的方法对用户视点进行精确预测。本文将使用 Almquist 等人^[25]在其工作中选用的一个典型的探索类 360 度视频，该视频可由 Youtube 上直接下载。在本文的实验中，原始 360 度视频仅运行 30 秒，即一个较短视频的持续时间。

为了更好覆盖用户视点的移动情况，本文使用静态的用户视点生成策略，即使用算法生成用户在不同时间点上的具体视点，而不使用用户在真实头戴设备下记录的真实视点。如图 5.1 所示，本文生成的用户视点的移动包含三个主要过程：横向快速移动、纵向慢速移动以及斜向移动。



图 5.1 用户视点移动轨迹示例

5.2 系统功能展示

本文实现了一个适用于 360 度视频的多路径传输系统,图 5.2 展示了一次传输过程完成后客户端生成的用户可观看的视频与原始 360 度视频的关系。其中,最左侧为等距柱状投影格式的原始 360 度视频,其上标记了两个不同时间段的用户视点,这两个用户视点对应的服务端所计算出的需要传输的视频块序列和用户真实视野分别以不同颜色的矩形进行标记。可以看到,系统中进行传输视频块的总和总是能够完全覆盖用户的真实视野区域,并存在一定的缓冲区域。因此,本文所实现的多路径传输系统具有一定的鲁棒性,即当用户真实视点小幅度偏移时也能保证最基础的视频流畅度。

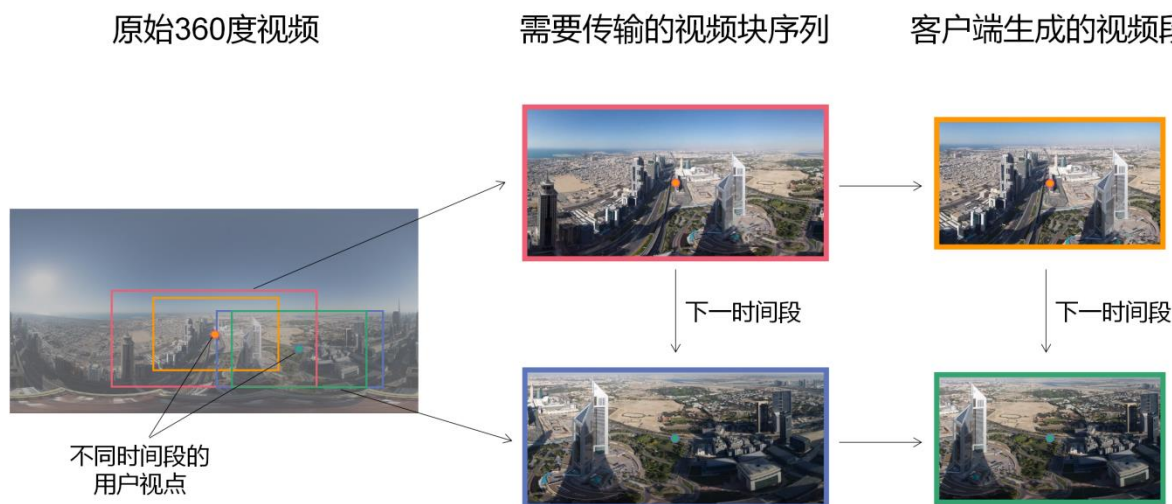


图 5.2 客户端生成视频与原始视频的关系

5.3 系统性能测试

本文提出了一种将 360 度视频分块分层并进行多路径传输的策略,期望通过此策略减少视频实时播放中的卡顿频率并提高用于传输用户视野内视频块的有效网络流量,从而提高用户体验质量。为了检验该策略相较于传统方法的优势所在,本文将设计实验测试不同的传输策略在不同网络带宽条件下的具体表现,统计卡顿频率和有效流量两个主要指标进行比较和分析。

为了提供不同的网络带宽环境,本文通过向物理链路上注入不同强度背景流量的方式调整每一虚拟用户所能够使用的网络带宽,而这些背景流量也可以由 360 度视频的传输这一应用层业务产生,因此,本文所设计的实验直接将系统中用户的数量作为一个主

要变量表征网络带宽环境的变化。具体而言,在实验中会创建若干个虚拟用户,每一个虚拟用户都持续与服务器之间进行 360 度视频的请求与应答过程,而系统将关注某一个特定虚拟用户的卡顿频率和有效流量指标。

需要进行比较的传输策略的选择上,由于本文提出的传输方法主要由两个维度构成:传输部分视频块、多路径传输,为了满足单一变量原则,本文将使用分别在这两个维度上有所改变的传输策略与本文所提出的方法进行比较。此外,由于期望获取 Wi-Fi 路径和 5G 路径的异同,实验中将单路径传输的策略在两条物理链路上分别进行实现和测试。因此,本文共使用四种不同的传输策略进行比较:a) 仅传输部分视频块、多路径传输(本文提出的方法),b) 仅传输部分视频块、Wi-Fi 单路径传输,c) 仅传输部分视频块、5G 单路径传输,d) 传输全部视频块、多路径传输。

实验所测得的不同传输策略下卡顿频率与系统中用户数量的关系如图 5.3 所示,由该图可以看出,传输所有块的策略在用户极少时也不能保证基本流畅度,因此基本不可用。而对于传输部分块的三种具体传输策略,卡顿频率都随系统中用户数量的增加呈上升趋势,其中两种单路径传输的策略在 3 个用户左右时已经达到近半的卡顿频率,用户体验质量较差。相比之下,本文所提出的多路径传输部分块的方法显著减缓了卡顿频率的上升趋势,在 4 个及以下用户时卡顿频率维持在较低水平上,保障了用户观看视频的基本流畅度。在用户数量较多时,单一用户所能够使用的网络带宽严重不足,所有传输方法都出现了较多次数的卡顿。

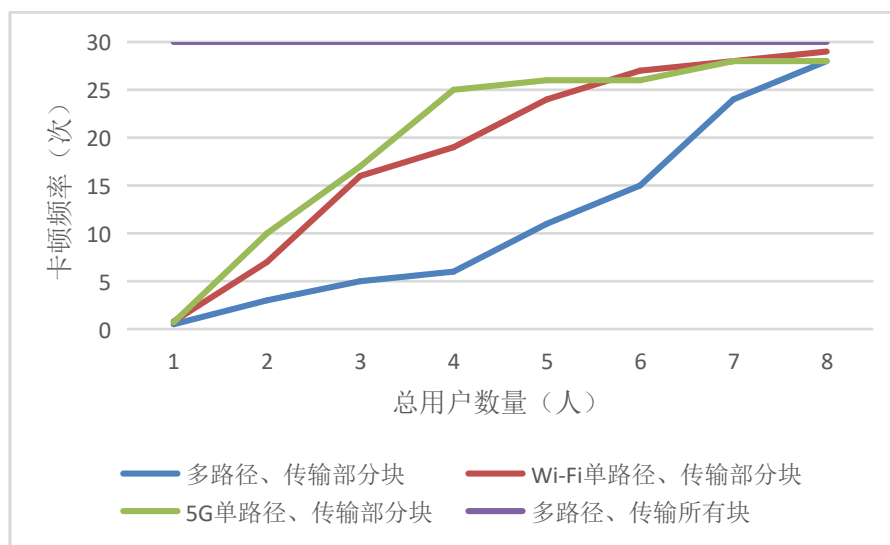


图 5.3 不同传输策略下单用户卡顿频率与系统用户数量的关系

实验所测得的不同传输策略下有效流量与系统中用户数量的关系如图 5.4 所示, 由该图可以看出, 对于所有传输策略, 有效流量都随系统中用户数量的增加呈下降趋势。其中, 传输部分块的三种策略在用户极少时都能完全满足传输需要传输的所有视频块的所有层的要求, 因而其有效流量在用户数量为 1 时相等, 而传输所有块的策略在用户极少时的有效流量已经显著低于以上三种策略, 随着用户数量的增加, 该方法的有效流量逐渐趋于 0, 因此基本不可用。而对于传输部分块的三种具体传输策略, 当用户数量小幅增加时, 有效流量迅速降低, 这是由于网络带宽不足以支持传输所需传输视频块的所有层而优先传输基础层和较低增强层导致的, 通过降低视频质量的方式避免卡顿, 这里损失的有效流量主要是较高增强层的流量。随着用户数量的进一步增加, 有效流量下降的趋势趋于平缓, 这说明此时流量已经基本上全部用于传输基础层, 而随着用户数量的进一步增加, 网络带宽环境更加恶化, 逐渐开始不能支持基础层的全部传输, 因而此时卡顿频率迅速上升。相较而言, 本文所提出的多路径传输部分块的方法显著减缓了有效流量的下降趋势, 在 4 个及以下用户时有效流量仍维持在较高水平上, 因此该方法在同等网络带宽条件下更好保证了用户视野内的视频质量, 从而保障了用户体验质量。

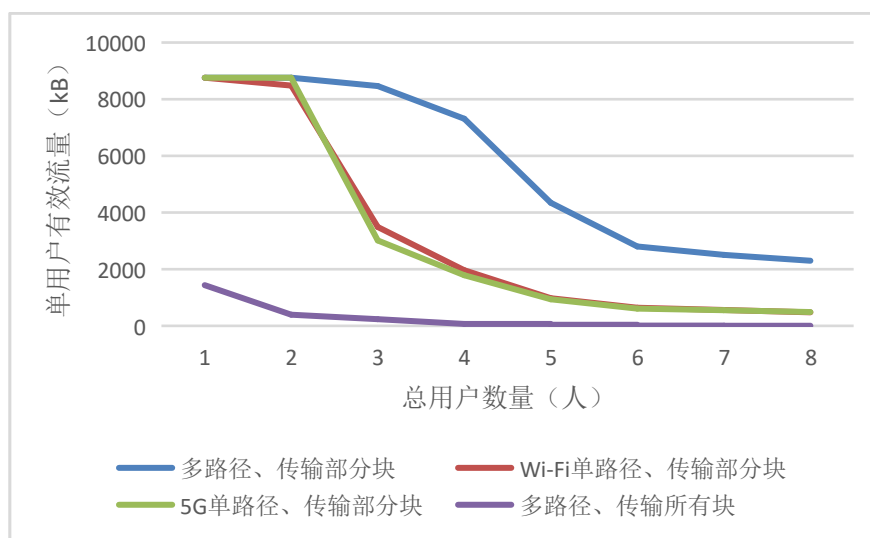


图 5.4 不同传输策略下单用户有效流量与系统用户数量的关系

综上所述, 对比实验表明, 本文所提出的将 360 度视频分块分层并进行多路径传输的方法能够显著降低用户观看过程中的卡顿频率并提高用于传输用户所观看的视频部分的有效流量, 从而提高了用户体验质量, 较好完成了本文的设计目标。



总结与展望

为了解决实时传输高分辨率的 360 度视频需要极高的网络带宽这一问题,本文提出了一种基于可伸缩视频编码的多路径传输机制。这一机制首先使用可伸缩视频编码将预先分块分段的 360 度视频分成不同质量的基础层与若干增强层,以文件的形式作为静态资源存放于服务器上。因而在具体传输过程中,可以利用 Wi-Fi 链路的稳定性传输各基础层以保障视频基本流畅度,利用 5G 链路的高带宽特性传输各增强层以提高用户观看的视频质量。在提出这一多路径传输机制的基础上,本文根据相关应用场景在物理设备上进行了实现。硬件层面上,搭建了 Wi-Fi 与 5G 两条物理链路并通过 IP 网段划分与静态路由配置等过程实现了全网互联互通。软件层面上,分别在多网卡服务器和单网卡 PC 上实现了基于 Django 的服务端和基于 Python 的客户端,从而基于 HTTP 协议实现了同时使用两条物理链路带宽进行 360 度视频传输的过程,从功能上满足了用户在客户端观看服务端上静态资源形式 360 度视频的需求。同时性能实验表明,在相同的网络带宽条件下,用户观看使用本文所提出多路径机制进行传输的 360 度视频过程中的卡顿频率相较于传统方法最多降低了 80%,而用于传输用户所真实观看视频部分的有效网络流量最多约是传统方法的 3 倍。这说明本文所提出的多路径传输机制能够显著提高用户体验质量。

在目前的实验中,本系统 5G 物理链路的平均下载速率较低,不符合本文对 5G 链路高带宽特点的认识,未来的工作可尝试定位并解决这一问题。在此基础上,未来的工作可探索更合理的视频切分方法。本文所设计的切分方法简单地将分辨率为 3840×1920 的 4K 原始 360 度视频按照 320×240 为单位分辨率进行切分,最终形成 12×8 个视频块。实验中可以发现,服务端决策的需要进行传输的视频块数量往往较多,其所覆盖的区域存在远大于用户真实视野的情况,因此网络中仍存在较多的无效流量。此外,可进一步优化视频传输机制。在本文所设计的传输机制中,每一时间间隔开始时都只对当前时间间隔内的各视频块各层进行决策,选取需要进行传输的部分。但如果将时间这一变量也作为此决策过程的输入,则在传输更远时刻的视频基础层以保证低卡顿频率和传输更近时刻的视频增强层以提高视频质量之间进行权衡将会是一个有趣的过程。



致谢

首先要感谢我的毕设导师，张珊老师。张老师对我在前期准备、中期实验以及后期成文的过程中遇到与提出的问题总是具有独到的见解，帮助我跳出了许多误区，更好地完成本文工作。

其次，感谢参与周一下午例行讨论会的所有同学，感谢你们对我所提出问题的积极解答。其中尤其要感谢胡彬彬学长，从繁重的课程负担中抽出时间与我分享论文，并对可伸缩视频编码等相关技术进行详细讲解。同时也要感谢彭冠涵同学在搭建和调试系统物理链路与设备上向我提供的帮助。



参考文献

- [1] Qian F, Han B, Xiao Q, et al. Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices[A]. Proceedings of the 24th Annual International Conference on Mobile Computing and Networking[C]. 2018: 99-114.
- [2] H. Bellini. The real deal with virtual and augmented reality[EB/OL]. <http://www.goldmansachs.com/our-thinking/pages/virtual-and-augmented-reality.html>, 2016.
- [3] Huawei. Whitepaper on the VR-Oriented Bearer Network Requirement[EB/OL]. <http://www-file.huawei.com/~media/CORPORATE/PDF/white%20paper/whitepaper-on-the-vr-oriented-bearer-network-requirement-en.pdf>, 2016.
- [4] Evgeny Kuzyakov. Under the hood: Building 360 video[EB/OL]. <https://code.facebook.com/posts/1638767863078802/under-the-hood-building-360-video/>, 2015.
- [5] Evgeny Kuzyakov. Next-generation video encoding techniques for 360 video and VR[EB/OL]. <https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/>, 2016.
- [6] Budagavi M, Furton J, Jin G, et al. 360 degrees video coding using region adaptive smoothing[A]. 2015 IEEE International Conference on Image Processing[C]. 2015: 750-754.
- [7] Duanmu F, Kurdoglu E, Liu Y, et al. View direction and bandwidth adaptive 360 degree video streaming using a two-tier system[A]. 2017 IEEE International Symposium on Circuits and Systems[C]. 2017: 1-4.
- [8] Mangiante S, Klas G, Navon A, et al. Vr is on the edge: How to deliver 360 videos in mobile networks[A]. Proceedings of the Workshop on Virtual Reality and Augmented Reality Network[C]. 2017: 30-35.
- [9] 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.



- [10]Ju R, He J, Sun F, et al. Ultra wide view based panoramic VR streaming[A]. Proceedings of the Workshop on Virtual Reality and Augmented Reality Network[C]. 2017: 19-23.
- [11]Corbillon X, Simon G, Devlic A, et al. Viewport-adaptive navigable 360-degree video delivery[A]. 2017 IEEE international conference on communications[C]. 2017: 1-7.
- [12]Sun L, Duanmu F, Liu Y, et al. Multi-path multi-tier 360-degree video streaming in 5G networks[A]. Proceedings of the 9th ACM Multimedia Systems Conference[C]. 2018: 162-173.
- [13]Ge X, Pan L, Li Q, et al. Multipath cooperative communications networks for augmented and virtual reality transmission[J]. IEEE Transactions on Multimedia, 2017, 19(10): 2345-2358.
- [14]Liu Y, Liu J, Argyriou A, et al. MEC-assisted panoramic VR video streaming over millimeter wave mobile networks[J]. IEEE Transactions on Multimedia, 2018, 21(5): 1302-1316.
- [15]Hou X, Dey S, Zhang J, et al. Predictive adaptive streaming to enable mobile 360-degree and VR experiences[J]. IEEE Transactions on Multimedia, 2020, 23: 716-731.
- [16]Bengio S, Vinyals O, Jaitly N, et al. Scheduled sampling for sequence prediction with recurrent neural networks[J]. Advances in neural information processing systems, 2015, 28.
- [17]Breiman L. Random forests[J]. Machine learning, 2001, 45(1): 5-32.
- [18]Amit Y, Geman D. Shape quantization and recognition with randomized trees[J]. Neural computation, 1997, 9(7): 1545-1588.
- [19]Kim T, Yue Y, Taylor S, et al. A decision tree framework for spatiotemporal sequence prediction[A]. Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining[C]. 2015: 577-586.
- [20]Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- [21]Liu J, Shahroudy A, Xu D, et al. Spatio-temporal lstm with trust gates for 3d human action recognition[A]. European conference on computer vision[C]. 2016: 816-833.



- [22]Mohamed A, Dahl G E, Hinton G. Acoustic modeling using deep belief networks[J]. IEEE transactions on audio, speech, and language processing, 2011, 20(1): 14-22.
- [23]Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. science, 2006, 313(5786): 504-507.
- [24]Alahi A, Goel K, Ramanathan V, et al. Social lstm: Human trajectory prediction in crowded spaces[A]. Proceedings of the IEEE conference on computer vision and pattern recognition[C]. 2016: 961-971.
- [25]Qian F, Ji L, Han B, et al. Optimizing 360 video delivery over cellular networks[A]. Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges[C]. 2016: 1-6.
- [26]Xu Y, Dong Y, Wu J, et al. Gaze prediction in dynamic 360 immersive videos[A]. proceedings of the IEEE Conference on Computer Vision and Pattern Recognition[C]. 2018: 5333-5342.
- [27]Almquist M, Almquist V, Krishnamoorthi V, et al. The prefetch aggressiveness tradeoff in 360 video streaming[A]. Proceedings of the 9th ACM Multimedia Systems Conference[C]. 2018: 258-269.