面向全息视频通信的自适应分块传输方法

朱原玮,黄亚坤,乔秀全*
(北京邮电大学网络与交换技术全国重点实验室,北京 100876)

摘 要: 基于分治和按需传输思想的分块传输技术是解决三维全息视频流传输的有效手段. 然而,现有的分块方案要么缺乏自适应机制,要么不适用于移动实时通信场景. 为此,本文提出了VVSTiler(Volumetric Video Streaming Tiling selector),一种面向全息视频通信的自适应分块传输方法,能够在动态且有限的计算和带宽资源下最大化视频的观感质量. 具体而言,本文对不同粒度的分块方案带来的影响进行了初步研究,发现细粒度的分块方案可提高动态网络资源的利用率,粗粒度的分块方案可保证视频编解码效率和鲁棒性. 基于此,本文构建了考虑预测视口、可用计算资源以及网络带宽等上下文信息的视频观感质量优化问题,并设计了一个高效的求解方案以支持在线的分块粒度决策. 本文在8iVFB(8i Voxelized Full Bodies)标准数据集上将VVSTiler与当前主流的分块传输方法进行了比较. 实验结果表明,VVSTiler在有偏差的视口预测情况下实现了高达60.4%的视频观感质量提升,在较准确的视口预测情况下平均每帧视频节省了27%的带宽资源.

关键词: 全息视频通信;体积视频流;点云视频;自适应分块;视口预测;视频观感质量
基金项目: 国家重点研发计划(No.2022YFB2902900);国家自然科学基金(No.62202065)
中图分类号: TP37;TP393.0 **文献标识码:** A **文章编号:** 0372-2112(2024)04-1144-11
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20230788

Towards Holographic Video Communications: An Adaptive Tiling Solution

ZHU Yuan-wei, HUANG Ya-kun, QIAO Xiu-quan*
(*State Key Laboratory of Networking and Switching Technology*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)

Abstract: Tile-based methods that use the divide-and-conquer and on-demand transmission techniques are promising to handle 3D holographic video streaming. However, the current solutions either lack an adaptive tiling scheme or cannot apply to mobile real-time scenarios. In this paper, we propose VVSTiler (Volumetric Video Streaming Tiling selector), an adaptive tiling selector for holographic video communications, which can adaptively maximize perceived video quality under dynamic and limited computing and bandwidth resources. To be specific, we first conduct a preliminary study on the impacts of different tiling schemes and find that fine-grained tiles improve the rational utilization of dynamic network resources and coarse tiles ensure coding efficiency and robustness, which stimulates us to construct an adaptive tiling optimization based on the predicted viewport, available computing resources, and network bandwidth; and then devise a fast algorithm to enable online tiling decisions. Rich experiments on the 8iVFB (8i Voxelized Full Bodies) datasets are conducted to compare VVSTiler with state-of-the-art tiling-based baselines. The results exhibit that VVSTiler can achieve up to 60.4% video quality improvements and save on average 27% bandwidth per frame against the closest competitor, in cases of terrible and accurate viewport predictions, respectively.

Key words: holographic video communications; volumetric video streaming; point cloud video; adaptive tiling; viewport prediction; perceived video quality

Foundation Item(s): National Key R&D Program of China (No.2022YFB2902900); National Natural Science Foundation of China (No.62202065)

收稿日期:2023-08-16;修回日期:2023-12-05;责任编辑:宋甲英
*通讯作者:乔秀全

1 引言

全息视频,又可称体积视频(volumetric video),是一种能够提供给用户六个自由度体验的新型沉浸式视频形式. 与具有三个自由度体验的360度全景视频和虚拟现实(Virtual Reality,VR)视频不同,全息视频增加了三个自由度的平移运动,增强了沉浸感和交互性[1]. 然而,全息视频数据量巨大,需要至少千兆比特每秒(Gbps)级别的网络带宽进行传输. 以具有代表性的三维点云视频格式为例,微软的Azure Kinect深度相机每帧采集了70.4 Mb的原始点云数据,在30 FPS(Frames Per Second)的采集帧率下相当于2.06 Gbps的数据量[2]. 当使用多个安放在不同角度的相机同步采集时,此数据量还会随着相机数量的增加而进一步增大. 因此,现有的移动网络环境很难支持全息视频流的实时通信.

现有的相关工作主要通过结合视频压缩[3~5]和自适应流[6~9]技术来实现全息视频的高效传输. 其中,基于分块的传输方法[6~14]已经在360度全景视频流和VR视频流中得到了广泛的应用,并展现出了许多优势,例如:(1)基于分块的传输方法能更好地兼容基于HTTP的动态自适应流标准(Dynamic Adaptive Streaming over Http,DASH),使视频块的质量等级(码率)适应动态的网络条件[6];(2)将分块技术与视口预测技术相结合,仅传输位于用户视口内的视频块,有效节省了带宽和计算资源. 因此,基于分块的传输方法逐渐被拓展到全息视频流通信领域.

目前,现有的全息体积视频分块传输方法可分为两类. 第一类固定或静态的分块模式. 代表性的方法有文献[2,7,8,11~14]根据经验采用固定的分块策略. 例如文献[2]将全息人物模型分割为2×3×2个立方体. 然而,当视频的内容从人物变为其他物体,或点的数量在数千至数百万之间变化时,固定的分块模式导致每个视频块在密集(稀疏)的场景中包含过多(过少)的点,造成带宽资源的浪费(传输效率的降低). 文献[9,10]预先评估分块粒度对视频编码效率造成的影响,然后选择一个合适的粒度进行分块. 然而,实际应用中移动网络和终端可用的计算资源易动态变化,这些静态的分块模式在动态环境下并不是自始至终的最佳选择. 第二类混合视觉显著性分块模式. 文献[15]提出混合视觉显著性分块方法,该方法能够准确地匹配用户视口,但需要对视频内容进行运动估计,计算开销较大,不适用于实时传输和电池容量有限的移动设备. 综上,现有的分块方法存在两个主要问题:一方面,在粒度上缺乏自适应机制,难以灵活应对不同场景的需求;另一方面,并不适用于移动实时通信,因此需要更高效和轻量化的解决方案.

本文提出一种面向全息视频通信的自适应分块传输方法(Volumetric Video Streaming Tiling selector,VVSTiler). 该方法根据预测的用户视口、终端可用的计算资源以及当前移动网络带宽情况自适应地选择最优的分块粒度;同时,本文设计了一个可在线决策的低时间复杂度的算法,无需理解视频的内容即可实现自适应粒度选择. 在8iVFB(8i Voxelized Full Bodies)标准数据集[16]和用户运动轨迹数据集[14]上进行了实验,将VVSTiler与其他代表性的分块方法进行了比较,结果表明VVSTiler在视频观感质量和传输效率两方面均有显著提升.

本文的主要贡献总结如下:

(1)动机实验表明粗粒度的分块策略可以容忍更大的视口误差,但也会导致传输更多视口外的点;与之相反,过于细粒度的分块策略可能导致视频内容的缺失,并且导致编解码的时间更长,验证了在全息视频流通信场景下动态地选择最优的分块策略的必要性.

(2)为了实现在移动全息视频流通信场景下动态地选择最优的分块策略,本文设计了一个自适应分块粒度选择器(VVSTiler),构建了一个考虑预测视口、终端可用的计算资源以及当前移动网络带宽情况的NP优化问题,并将其转化为了一个低时间复杂度问题,无需理解视频内容即可实时求解.

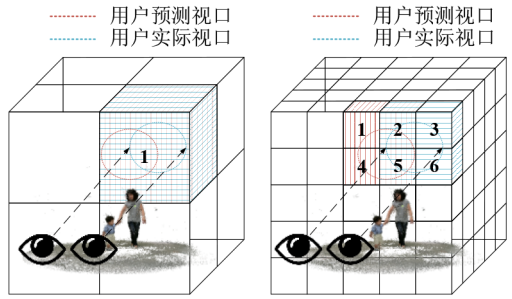
(3)本文搭建了移动全息视频流通信的原型系统,并定性和定量地对VVSTiler进行了评估. 结果表明,VVSTiler的传输性能明显优于其他对比方法,在峰值信噪比指标上比目前最好的方法提高了60.4%,在平均传输每帧视频所需带宽上节省了27%.

2 研究动机

2.1 分块粒度对视口预测的容错性的影响

图1给出一帧点云表示的全息视频示例,用户预测视口(红色圆圈区域)和实际视口(蓝色圆圈区域)存在偏差. 图1(*a*)和图1(*b*)分别为使用2×2×2和5×5×5两种分块策略的情形. 在5×5×5这种较细粒度的分块下,第3个和第6个视频块在用户实际视口,但不在预测视口,因此会被分配较低的质量等级甚至在传输前被丢弃,这种内容缺失导致的视频观感质量下降比分辨率降低或失真更严重. 此外,第1和第4个视频块在用户预测视口,但不在实际视口,因此会被传输甚至被分配较高的质量等级,造成带宽和计算资源的浪费. 但若此时采用2×2×2的分块策略,在用户预测视口内的视频块(第1块)将正好处于实际视口. 该示例定性说明了粗粒度的分块策略可以容忍更大的视口预测误差,但更多视口外的点的传输也浪费了更多的带宽资源. 这些结果验证了根据实际视口误差和网络状况动态选择最

优分块策略的必要性.



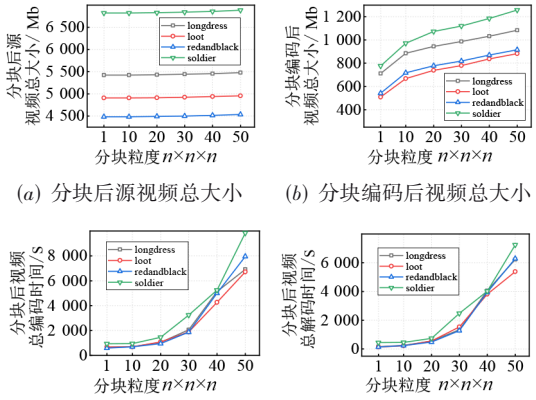
(a) 2×2×2分块策略 (b) 5×5×5分块策略

图1 两种不同分块粒度对视口预测容错性的影响示意图

2.2 分块粒度对分块开销和编解码效率的影响

本实验以8iVFB标准数据集[16]中的四个点云视频为例对其进行分块处理,视频名称分别为longdress,loot,blackandred和soldier. 实验在配备了48核Intel(R) Xeon(R) Gold 5118 CPU @2.30 GHz的边缘服务器上进行. 实验采用从10×10×10到50×50×50五种不同粒度的分块策略,此外还采用了1×1×1的分块策略,指不进行任何分块处理的源视频. 值得注意的是,分块策略与点云视频编解码器相互独立. 在本文中,本文采用Draco[4]对每个视频块进行编码和解码,因为,与PCL(Point Cloud Library)[3]和LEPCC(Limited Error Point Cloud Compression)[17]相比,Draco可以在更短的编码时间内实现更高的无损压缩比,同时保证解压缩后视频的质量[9].

由图2(*a*)可知,视频划分越细,分块后源视频总大小越大,因为每个视频块都需要包含一些头部信息,如视频块的索引和点数等. 然而,这些信息所占空间较小,所以增长幅度不大. 由图2(*b*)可知,视频划分越细,分割开销越大,分割开销指分块编码后视频总大小与分块后源视频总大小之比[9]. 由图2(*c*)和图2(*d*)可知,视频划分越细,分块后视频总编解码时间越长,这是因为将视频帧分割成更小的视频块后,编解码器需要对点云进行多次编码和解码操作,降低了编解码的效率. 以一个极端的情况为例做解释,当每个视频块都只有一个点时,对一个独立的点编码8次明显比在基于八叉树[18]或kd树[19]这种数据结构的压缩机制下一次性编码8个点的集合效率低,这是因为视频块之间的关联信息没有得到充分利用[20]. 值得注意的是,当分块粒度大于20×20×20(即图2中的n>20)时,编码和解码效率会显著下降. 此外,更多的视频块意味着产生更多的HTTP接口和解码任务,会进一步增加计算能力较低的移动设备的负担. 因此在实际应用中,需要谨慎考虑是否采用过于细粒度的分块策略.



(a) 分块后源视频总大小 (b) 分块编码后视频总大小

(c) 分块后视频总编码时间 (d)分块后视频总解码时间

图2 不同粒度的分块策略对分块开销和编解码效率的影响

3 相关工作

3.1 基于分块传输的360度视频流通信

使用分块技术与自适应流技术优化用户的体验质量已广泛存在于360度全景视频领域[21,22]. Xie等人[21]使用了一种基于概率的方法,提前获取视频块以弥补视口预测误差,并设计了一个用户体验质量驱动的自适应流系统. Yadav等人[22]将比特率分配问题建模为关于视口和缓冲区占用的利润函数的多类背包问题. 以上方法采用的是固定的分块策略. 一些研究工作提出了自适应的分块策略[20,23]. Xiao等人[23]提出了OpTile,通过估算存储成本,将问题建模为整数线性规划以求解最优分块方案. Zhang等人[20]分析了分块对解码时间的影响,并将自适应分块机制引入至自适应比特率算法中. 总之,上述方法在360度全景视频流领域为提高用户体验质量做出了重要贡献,但由于视频格式不同,无法直接应用于三维全息视频,设计适用于体积视频格式的自适应分块方案面临着额外的挑战.

3.2 基于分块传输的体积视频流通信

受360度视频流分块传输的启发,一些体积视频流系统将点云帧空间划分为多个视频块,并分配合适的质量等级,以提高网络资源的利用率[6~14]. 然而,这些系统主要侧重于比特率的自适应和视频块之间的资源分配,如引入贪心算法[12]、启发式算法[7]、体验质量驱动方案[2,11]、多选择背包问题[8]等. 它们通常采用固定的分块策略,缺乏深入研究不同分块粒度对传输性能的影响. 此外,Han等人[9]将视频分割成三种大小的视频块,量化分析了分块开销. Lee等人[10]利用PD树的层次结构来处理场景中的密度变化. Subramanyam等人[14]提出了一种低时间复杂度的分块方法,将整个点云帧空间分成4部分. Li等人[15]提出了一种混合视觉显著性和分层聚类的分块方案,以更好地匹配用户视口. 综上所述,虽然这些方法探讨了固定分块策略的不足之

处,或设计了新的分块方法,但它们都忽略了分块粒度对视口预测误差的容忍度,缺乏对动态视频内容的通用性和移动实时场景的适用性.

4　全息视频流系统

图3概述了本文提出的移动全息视频流通信系统,主要包括以下三个过程.

(1)服务器端的视频分块和编码. 首先,服务器采集来自多个角度的深度相机捕获的原始点云视频流,并进行同步和拼接. 与现有的固定分块策略不同,本文在服务器端部署了一个自适应分块选择器. 该选择器根据预测的用户视口、终端可用的计算资源以及当前移动网络带宽情况选择最优的分块粒度. 然后,它按照此粒度将当前一个时间段内的视频帧在空间上分割成视频块,并剔除在预测视口之外的块,降低视频传输的数据量. 最后,每个视频块被编码至合适的质量等级.

(2)无线传输. 使用基站传输编码后的视频块,去除有线连接,提供不受束缚的环境,使用户更自由地探索三维空间.

(3)客户端的视频解码播放. 用户佩戴增强现实(Augmented Reality,AR)眼镜与沉浸式场景进行交互. 客户端首先将接收到的码流解码成视频块,使用排序器和融合器将它们拼接成完整的视频. 然后重建的视频被渲染和缓冲,供播放器播放. 同时,客户端在播放每帧视频时向服务器报告预测的用户视口.

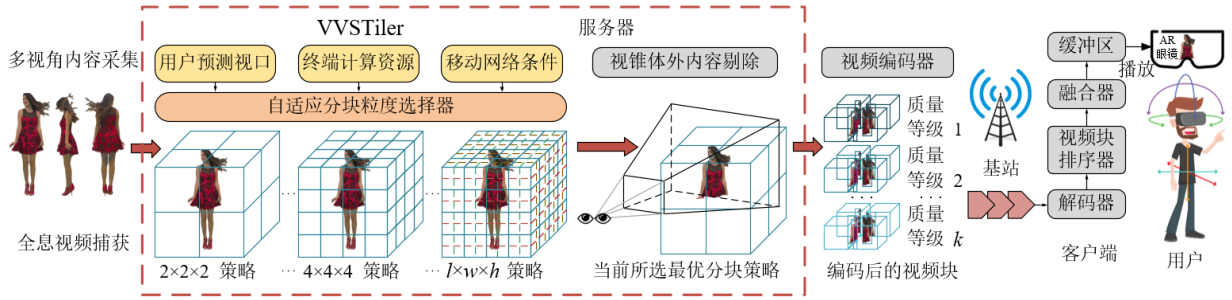


图3　VVSTiler系统概述图

5　VVSTiler详细设计

5.1　问题描述

与现有的采用固定分块模式的全息视频流通信系统相比,VVSTiler可适应不同程度的用户视口预测误差、终端可用的计算资源、当前移动网络带宽情况以及视频内容. 假设在一个一般的全息视频流通信系统中,用户可以不受任何限制地改变头部旋转方向和身体平移运动,定义$S=\{s_1,s_2,\cdots,s_m\}$为该全息视频可用的分块策略的候选集. 对于任意的分块策略$s_i\in S$,令l_i,w_i和h_i分别表示整个三维空间在长度、宽度和高度三种尺度上被分成多少块长方体. 令$|s_i|$表示视频块的总数,则$|s_i|=l_i\times w_i\times h_i$. 本文目的是在传输一组视频帧(Group Of Frames,GOF)前确定最优的分块策略s_{opt},将其用于后续该GOF的分割.

解码时间:对于任意一种编解码器,解码一帧点云视频所需的时间和计算资源与该帧视频包含的总点数、压缩等级以及块数$|s_i|$有关. 假设解码一个最低压缩等级的最小视频块(即一个体素)所需的计算资源为R_{unit},通过定义一个参数$\zeta_{g,p,k}$,可使解码第g帧视频中压缩等级为k的视频块p所需的计算资源表示为$R_{g,p,k}=f_1(R_{\text{unit}};\zeta_{g,p,k})$,可通过将该变量归一化为中央处理器的每秒浮点数运算数来描述所需的时间. 令R_{core}表示终端设备的单核处理器在一个GOF时间内可以提供的计算资源,N_{core}表示核心数,R_{total}表示总计算资源,则$R_{\text{total}}=N_{\text{core}}\times e\times R_{\text{core}}$,其中,$e$为在多核工作模式下执行解码的转换效率.

定义1　使用分块策略s_i对第G个GOF分块后视频的解码时间DT定义:

$$\text{DT}_G(s_i)=\frac{\sum_g\sum_p\sum_k R_{g,p,k}\times x_{g,p}\times z_{g,p,k}}{N_{\text{core}}\times e\times R_{\text{core}}}\tag{1}$$

其中,$g\in G$;$p\in P$;$k\in K$;P为视频块集合;K为可使用的压缩等级集合. $x_{g,p}$为二进制变量,表示视频块p是否在用户视口内,$x_{g,p}=1$表示p在视口内,应该被发送至客户端;$x_{g,p}=0$表示p不在视口. $z_{g,p,k}$为二进制变量,表示是否发送压缩等级为k的视频块p,$z_{g,p,k}=1$表示发送,$z_{g,p,k}=0$表示不发送.

传输时间:同样地,对于任意一种编解码器,编码一帧被分块后的点云视频所产生的数据量与该帧视频包含的总点数、压缩等级以及块数$|s_i|$有关. 假设编码一个最低压缩等级的最小视频块(即一个体素)后所产生的数据量为D_{unit},通过定义一个参数$\delta_{g,p,k}$,可使编码第g帧视频中压缩等级为k的视频块p后所产生的数据量表示为$D_{g,p,k}=f_2(D_{\text{unit}};\delta_{g,p,k})$,具体函数表示依赖所用的编解码器中压缩算法的原理.

定义 2 使用分块策略 s_i 对第 G 个 GOF 分块后视频的传输时间 TT 定义：

$$\mathrm{TT}_G(s_i)=\frac{\sum_g\sum_p\sum_k D_{g,p,k}\times x_{g,p}\times z_{g,p,k}}{\mathrm{BW}_G} \tag{2}$$

其中，BW_G 为第 G 个 GOF 期间网络的平均带宽.

编码时间：编码效率也是全息视频通信中的一项关键因素. 然而，编码通常在高性能的多媒体服务器上完成，解码通常在 AR 眼镜等资源有限的移动设备上完成，因此，解码效率是整个通信系统的性能瓶颈. 考虑到编码时间的定义与定义 1 相似，这里不再对编码时间进行冗余的形式化描述.

视频观感质量：从直观上分析，用户对视频的观感质量与视频内容的完整性和质量等级有关. 一方面，用户实际观看到的视频块必须与预测视口匹配，否则，用户看到的视频将会缺失内容，导致不良的观看体验. 另一方面，用户通常偏好高分辨率的视频内容. 因此，对于第 g 帧中的某个视频块 p，其对视频观感质量的贡献与 p 包含的点数和 p 是否在用户视口内有关. 将这一贡献表示为

$$C_{g,p}=\sum_k k\times\rho_{g,p}\times x_{g,p}\times z_{g,p,k} \tag{3}$$

其中，$\rho_{g,p}$ 表示第 g 帧中视频块 p 的密度权重，其被定义为视频块 p 包含的点数与用户视口内的总点数的比值，计算公式如下：

$$\rho_{g,p}=\frac{N_{g,p}}{\sum_p N_{g,p}} \tag{4}$$

其中，$N_{g,p}$ 为视频块 p 包含的点数.

定义 3 使用分块策略 s_i 对第 G 个包含 $|G|$ 帧的 GOF 分块后视频的观感质量定义：

$$Q_G(s_i)=\log\left(\frac{\sum_g\sum_p C_{g,p}}{\sum_g\sum_p |K|\times\rho_{g,p}\times x_{g,p}}\right) \tag{5}$$

基于上述内容，我们的目的是最大化 $Q_G(s_i)$.

5.2 问题证明

为在有限的计算和带宽资源下实时地传输最高观感质量的全息视频，进行以下优化：

$$\begin{aligned}\mathrm{P}_1:\ & s_{\mathrm{opt}}=\mathop{\arg\max}_{s_i,z_{g,p,k}} Q_G(s_i)\\ \text{s.t. }& \mathrm{DT}_G(s_i)\leqslant u_d\\ & \mathrm{TT}_G(s_i)\leqslant u_t\\ & \mathrm{ET}_G(s_i)\leqslant u_e\\ & \sum_{k=1} z_{g,p,k}=1,\forall g\in G,p\in P\end{aligned} \tag{6}$$

其中，u_d,u_t,u_e 分别表示解码时间 DT、传输时间 TT、编码时间 ET 的最大上界，用来保证视频通信的实时性. 例如，如果要使整个通信中每个流程的帧率均达到 30 FPS，则这些上界应满足 $u_d=30\times|G|,u_t=30\times|G|,u_e=30\times|G|$.

即使将解空间限制在一个有限集合 $s_i\in S$，也仍然很难从三维空间中实时找到问题 P_1 的最优解 s_{opt}. 接下来，证明 P_1 属于 NP 难问题.

引理 1 多重选择背包问题是 NP 难问题.

证明 给定 n 类互不关联的物品 $\{I_1,I_2,\cdots,I_n\}$，将其放入一个容量为 c 的背包中. 每个物品 $j\in I_i$ 有重量 $w_{i,j}$ 和价值 $v_{i,j}$. 问题的目标是在不超过背包容量 c 的条件下，从每个类中选择一个物品，使背包的总价值最大. 引入一个二进制变量 $y_{i,j},y_{i,j}=1$ 表示在第 I_i 类中选择了物品 j，则多重选择背包问题可表述为

$$\begin{aligned}&\max\sum_{i=1}^{n}\sum_{j\in I_i} v_{i,j}y_{i,j}\\ &\text{s.t.}\quad \sum_{i=1}^{n}\sum_{j\in I_i} w_{i,j}y_{i,j}\leqslant c\\ &\sum_{j\in I_i} y_{i,j}=1,\ i=1,2,\cdots,n\\ &y_{i,j}\in\{0,1\},\ i=1,2,\cdots,n,\ j\in I_i\end{aligned} \tag{7}$$

其中，文献[24]具体证明了问题(7)是 NP 难的. 证毕.

推论 1 问题 P_1 是 NP 难问题.

证明 将问题 P_1 中的每个变量映射到多重选择背包问题中对应的变量，问题 P_1 可转化为以下问题：(1)给定 $|s_i|$ 类互不关联的物品 $\{p_1,p_2,\cdots,p_{|s_i|}\}$，即质量等级，将其放入一个容量为 $\min\{u_d,u_t,u_e\}$ 的背包中；(2)每个物品(质量等级)$k\in p_i$ 有价值 $C_{g,p}$ 和重量($\min\{u_d,u_t,u_e\}$ 对应的 $R_{g,p,k}$ 或 $D_{g,p,k}$)；(3)问题的目标是在不超过解码时间、传输时间以及编码时间的最大上界的条件下，从每个质量等级中选择一个视频块，使总视频观感质量最大；(4)引入一个二进制变量 $z_{g,p,k}$，$z_{g,p,k}=1$ 表示在质量等级 k 中选择了视频块 p. 因此，当给定一个特定的分块策略后，我们可以将问题 P_1 的一种简化版本转化为对应的多重选择背包问题，因此问题 P_1 是 NP 难的. 证毕.

5.3 问题求解

虽然多重选择背包问题在实际中存在一些有效的解决方案[25]，但问题 P_1 的简化版本已经是 NP 难，在 NP 难问题的上层再增加一层决策后，即分块粒度的决策，将使问题 P_1 变得更难求解. 考虑到算法的复杂度和移动实时通信需求，本文设计了一个易于计算的求解方案，实现快速自适应分块.

首先引入两个变量，缺失率(Miss Ratio，MR)和浪费率(Waste Ratio，WR)[20]，从相反的角度最大化视频观感质量. MR 表示未被传输的视频块中有多少点在用

户实际视口中,WR表示将被传输的视频块中有多少点不在用户实际视口中,计算公式分别如下:

$$\mathrm{MR}(s_i)=\frac{M_{\mathrm{miss}}}{M_{\mathrm{FoV}}} \tag{8}$$

$$\mathrm{WR}(s_i)=\frac{M_{\mathrm{wast}}}{M_{\mathrm{FoV}}} \tag{9}$$

其中,M_{miss}表示与实际视口相比,预测视口中缺失的点的数量;M_{wast}表示与实际视口相比,预测视口中浪费的点的数量;M_{FoV}表示实际视口中点的总数. 为更清晰地描述每个变量的含义,图4给出了一个二维投影后的示例. 在该示例中,MR和WR分别等于0.7和0.6.

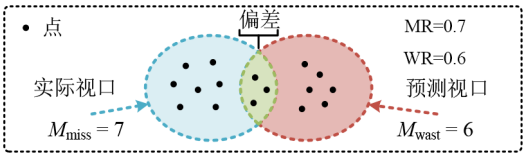


图4　缺失率MR和浪费率WR的计算示例图

显然,MR和WR的值越小,用户观看到的视频内容越完整,感受到的视频质量越好,而且传输的数据量越小. 因此可以推断出,分块粒度越细,MR越大,WR越小;反之MR越小,WR越大. 在此基础上,我们设计了一个视频观感质量惩罚指标对分块策略s_i进行评价:

$$\hat{Q}_G(s_i)=\beta\cdot\mathrm{MR}(s_i)+(1-\beta)\cdot\mathrm{WR}(s_i) \tag{10}$$

其中,β是两个变量之间的权重,其值根据实际环境调整.

通过同时最小化MR和WR,可以将视频观感质量$Q_G(s_i)$的最大化转化为视频观感质量惩罚$\hat{Q}_G(s_i)$的最小化,将问题P_1简化为以下:

$$\mathrm{P}_2:\arg\min_{s_i}\hat{Q}_G(s_i) \tag{11}$$

问题P_2的约束条件与问题P_1中的不等式约束相同. 通过求解问题P_2,可达到最大化视频观感质量和最小化传输数据量,同时保证编解码效率的目的,在考虑每种影响因素的情况下自适应地为全息视频流通信选择最合适的分块粒度. 对于问题P_2,本文设计了算法1在多项式时间内快速求解. 具体而言,首先根据问题P_2的约束条件分别求解符合约束的分块策略集合(步骤1~3),并对三个集合求交集,得到符合视频通信实时性要求的候选集合(步骤4);如果该候选集合不为空,则根据式(10)计算集合中每个分块策略的视频观感质量惩罚(步骤5~8),对所有分块策略按照惩罚排序后,输出最小惩罚对应的分块策略(步骤9,10,11,16). 如果该候选集合为空,则对根据式(1)求解的符合约束的分块策略候选集合中的所有分块策略按照解码时间排序后,输出最小解码时间对应的分块策略(步骤12~16). 当输入集合S为有限集时,算法1的时间复杂度为$O(N)$,因此可在多项式时间内快速求解.

算法1　自适应分块粒度在线决策算法

输入:分块粒度候选集$S=\{s_1,s_2,\cdots,s_m\}$,计算资源R_{core},网络带宽BW_G,上界u_d,u_t,u_e

输出:当前最优分块策略s_{opt}

1. $S_d\leftarrow\{s_i|\mathrm{DT}_G(s_i)\leqslant u_d\}$;
   /*根据式(1)求解符合约束的分块策略候选集
2. $S_t\leftarrow\{s_i|\mathrm{TT}_G(s_i)\leqslant u_t\}$;
   /*根据式(2)求解符合约束的分块策略候选集
3. $S_e\leftarrow\{s_i|\mathrm{ET}_G(s_i)\leqslant u_t\}$;
   /*类比式(1)求解符合约束的分块策略候选集
4. $S_{\mathrm{can}}=S_d\cap S_t\cap S_e$;
   /*对三个分块策略候选集合求交集
5. IF $S_{\mathrm{can}}\neq\varnothing$ THEN
6. FOR $s\in S_{\mathrm{can}}$ DO
7. 计算视频观感质量惩罚$\hat{Q}_G(s_i)$;
8. END FOR
9. $\{s_1,\cdots,s_i,s_j,\cdots|\hat{Q}_G(s_i)<\hat{Q}_G(s_j)\}\leftarrow\mathrm{Sort}(S_{\mathrm{can}})$;
   /*对集合S_{can}从小到大排序
10. $s_{\mathrm{opt}}=\min(S_{\mathrm{can}})$;
    /*最小视频观感质量惩罚对应的分块策略
11. END IF
12. ELSE
13. $\{s_1,\cdots,s_i,s_j,\cdots|\mathrm{DT}_G(s_i)<\mathrm{DT}_G(s_j)\}\leftarrow\mathrm{Sort}(S_d)$;
    /*对集合S_d从小到大排序
14. $s_{\mathrm{opt}}=\min(S_d)$;
    /*最小解码时间对应的分块策略
15. END ELSE
16. 返回s_{opt}.

5.4　计算复杂度分析

为具体解释VVSTiler一次最优分块粒度决策的计算复杂度,本文设$O(1)$为判断一个点或一个立方体是否在一个视锥体内的时间复杂度[26]. VVSTiler的决策过程包括三部分:(1)视频分割的时间复杂度$O(N)$,其中,N为点云视频帧中点的总数;(2)根据MR和WR的定义,计算一个分块策略$s_i=l_i\times w_i\times h_i$对应的两个变量的时间复杂度为$O(l_iw_ih_i+l_iw_ih_i+l_iw_ih_in_p)$,包括了判断视频块$p$是否在用户预测视口(视锥体)内,判断视频块$p$是否在用户实际视口(视锥体)内,以及判断视频块$p$中的每个点是否在用户预测和实际视口(视锥体)内,其中,n_p为视频块p中点的总数;(3)对于所有的候选分块策略,总时间复杂度为$O(|S|(N+(|s_i|(2+n_p))))$. 综上,当S为有限集时,VVSTiler的总时间复杂度近似为$O(N)$.

本文计算出,当处理8iVFB标准数据集级别的点云视频时(大约每帧80万个点),在主流的桌面级媒体服

务器上运行VVSTiler一次的平均时间约为20 ms,满足实时通信(30 FPS)的需求.

6 实验

在本节中,首先搭建了如图3所示的全息视频流通信原型系统,然后将VVSTiler与3个典型的分块方法进行了定性和定量比较.

6.1 数据集

(1)全息视频标准数据集. 实验采用由MPEG提供的8iVFB标准数据集[16],包括四个动态的点云序列:longdress,loot,blackandred和soldier. 每个序列以30 FPS的帧率采集了10 s的全息三维空间,分辨率为1 024×1 024×1 024个体素.

(2)用户运动轨迹数据集. 实验采用文献[14]收集的六自由度的用户运动轨迹数据,该数据集记录了26名用户观看Unity引擎渲染的8iVFB动态点云内容时的头部运动轨迹. 每个用户被要求佩戴Oculus Rift头戴显示器来观看每个视频,然后记录他们在观看每帧视频时的平移位置和旋转方向.

6.2 基准方法

最近360度视频研究领域提出了一些自适应分块方法,如文献[20,23],这些方法是专门为具有三个自由度体验的基于像素的二维视频设计的. 由于六自由度的全息视频在数据格式、空间维度和自由度上与360度视频完全不同,因此无法与360度视频领域中的分块方法直接进行比较. 同时,全息视频流领域自适应分块方法较少. 为评估第一种自适应分块方法VVSTiler,本文选择文献[2]和文献[11]采用的两种代表性的分块策略进行比较. 这两种方法通过计算8iVFB视频中点云人物的最小包围盒,根据经验使用固定且合适的分块粒度. 此外,本文还将VVSTiler与未分割的源视频1×1×1进行比较,记为"monoblock",此方法具有最高的视频观感质量,但传输数据量最大. 此外,Subramanyam等人[14]提出了一种低复杂度的分块方法,通过在XZ平面上为放置在物体周围的四个虚拟相机分配点云来对物体进行空间划分. 为提高压缩效率和降低数据量,该方法未考虑动态上下文资源将视频块的数量限制为4个. 由于作者没有提供源码和足够的信息以供复现,且考虑到该方法中的分块策略与文献[2]和文献[11]相似,因此本文最终选择文献[2]和文献[11]代替文献[14]作为代表性的基准对比方法.

6.3 实验设置

本文搭建了VVSTiler在全息视频通信应用中的原型系统,如图5所示. 为了实现这一系统,在不同角度上放置了12个微软Azure Kinect深度相机,用以采集场景中心的人物或物体,通过PCL库工具箱[3]融合12个视角的点云视频流. 此外,本文使用C/C++实现了VVSTiler并部署在多媒体服务器上,用来自适应地选择分块方案、分割和编码视频. 然后,我们使用Aruba WiFi路由器通过无线网络将编码后的比特流传输到HoloLens 2眼镜上,并使用Unity引擎渲染解码后的全息视频块. 最后,用户可以自由地观看渲染和缓存好的视频,获得六自由度的沉浸式体验.

图5 VVSTiler全息视频流通信原型系统

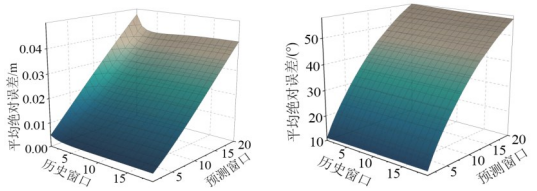
本工作的重点在于对自适应分块选择器模块的验证,与所选用的视频编解码器和压缩质量等级无关. 考虑到Draco可以在更短的编码时间内获得更高的无损压缩比,同时保证解压缩后视频的质量[9],我们选用Draco作为系统中的编解码方案,具体压缩等级(-cl)参数设置为7,量化位数(-qp)设置为11,均为Draco中的默认参数设置. 计算资源单位R_{unit}设置为1,视口内容剔除操作采用Lighthouse3d库[27]中的视锥体剔除函数. 为评估系统的通用性,我们采用现实采集的网络带宽轨迹数据集[28]模拟移动网络环境(0~100 Mbps). 其他参数如u_d,u_t,u_e被设置为保证整个传输帧率达到30 FPS. 后续实验在一台搭载了6核AMD Ryzen 5 2600X CPU @3.60 GHz和16 GB内存的桌面级服务器上进行.本文使用Unity 2020.3.28f1c1版本渲染点云帧,其中每个点的坐标缩放尺度为0.008单位的固定的偏移量,以对应用户运动轨迹数据集中的每个用户视角.

6.4 视口预测分析

全息视频的视口预测与三自由度的360度视频相比更具有挑战性,因为用户的平移运动坐标(X,Y,Z)也需要预测. 本文采用了两种视口预测模型:线性回归(Linear Regression,LR)和带有一个隐藏层的多层感知器神经网络(MultiLayer Perceptron,MLP). 选择这两种模型的原因有以下两点:(1)LR和MLP分别是线性模型和非线性模型中最典型和最简单的代表,这种选择确保了预测模型的轻量性和实时性;(2)考虑到移动设备的计算资源和电池容量有限,一些基于深度学习的方法,如长短期记忆网络[29,30],尽管可以提高一定的预测精度,但其高计算成本使其不适用于此场景.

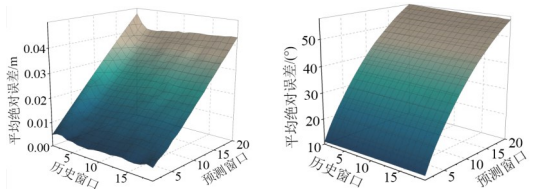
根据文献[9]的分析,使用单一的模型直接预测六个维度的视口坐标过于复杂. 因此,我们分别预测每个维度,然后结合每个维度的预测结果得出最终的六自由度视口. LR和MLP在这种策略下可获得较好的预测

精度. 我们从用户运动轨迹数据集中随机选择80%的轨迹用于训练LR和MLP模型,使用剩余的20%进行测试,测试结果如图6所示. 实验过程中,假设当前播放的视频帧为T,我们将包含从$T-h_1$到T播放的帧的历史窗口中的运动轨迹作为LR和MLP模型的输入,将在$T+h_2$播放的帧的预测窗口中的运动轨迹作为样本标签,其中h_1和h_2分别为历史窗口和预测窗口的长度. 我们采用预测视口与实际视口之间的平均绝对误差(Mean Absolute Error,MAE)衡量精度. MAE越小表示精度越高.



(a) LR在平移维度的结果

(b) LR在旋转维度的结果



(c) MLP在平移维度的结果

(d) MLP在旋转维度的结果

图6　LR和MLP分别在平移和旋转维度的预测结果

6.5　分块频率分析

自适应分块选择器在全息视频流通信中的每一帧都可以执行. 然而,以每帧一次的频率来求解最优的分块粒度会带来巨大的时间开销,在实时场景中不切实际,也不必要. 通常情况下相邻两帧之间的视频内容差异并不显著. 因此,我们进行实验来研究分块频率对决策时间和视频观感质量的影响.

我们将一组视频帧(GOF)中的帧数从1逐渐增加到20,并在传输每个GOF之前执行最优分块粒度的决策. 从图7的结果可看出,随着频率的降低(即GOF中帧数的增加),分块粒度决策的总时间减少,但视频观感质量惩罚增加,意味着视频观感质量下降. 综合考虑时间成本和视频观感质量,我们最终选择每10帧执行一次最优分块粒度决策,在时间开销和视频质量之间取得平衡.

6.6　分块粒度对MR和WR影响

本节研究不同分块粒度和两个所提指标(MR和WR)之间的关系,以验证两个指标的合理性. 在实验中,我们使用七个不同粒度的分块策略对测试视频进行分割,范围从1×1×1到7×7×7. 从图8可看出,视频划分越细,MR越大,WR越小. 其中,WR的值较大,在

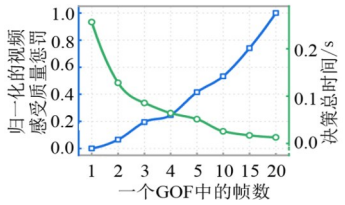


图7　分块选择频率分析

0.42和0.58之间,这是因为VVSTiler只要判断出一个视频块与用户的视锥体相交,就会选择传输该视频块. 从结果来看,MR和WR之间的这种矛盾关系启发了我们可以通过最小化两个指标的方式自适应地选择最优的分块策略,平衡视频观感质量和传输效率,从另一种角度避免了复杂问题的优化求解.

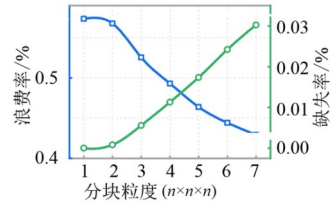


图8　分块粒度对MR和WR影响

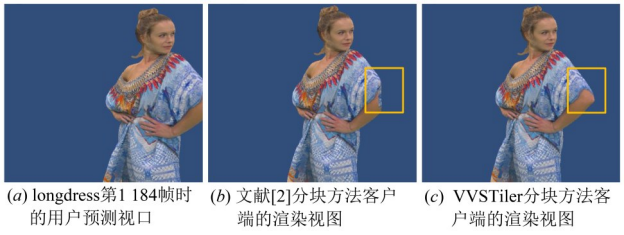
6.7　VVSTiler与其他方法的对比

6.7.1　主观定性比较

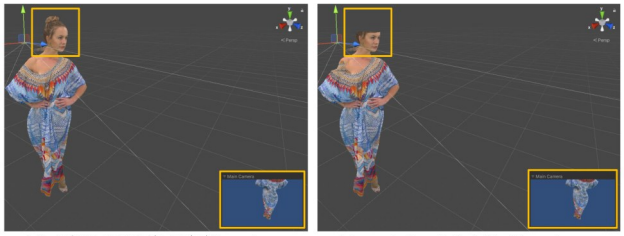
为直观理解VVSTiler相比其他分块方法带来的性能增益,图9中展示了主观比较实验的两种情况.

(1)图9(a)展示了longdress视频内容在第1 184帧时的用户预测视口,其中人物的肘部不在预测视口内. 当使用文献[2]中的分块策略时,肘部所在的视频块未被传输至客户端. 然而,用户的实际视口如图9(b)所示,因此,用户只能看到缺失肘部的人物形象,产生不良的观看体验. VVSTiler在这种情况下选择了一个较粗糙的分块策略(1×5×2). 其中,人物肘部与她的身体被分在同一个视频块中. 由于身体在用户预测视口内,所以连带着肘部的视频块一起被传输至了客户端,用户最终看到的视图如图9(c)所示. 该结果表明,当存在视口偏差时,VVSTiler可通过自适应地选择最优分块策略以容忍视口偏差,从而实现更加鲁棒的视频观感质量.

图9(d)和9(e)展示了longdress视频内容在第1 178帧的一个示例. 图的右下角显示了用户的实际视口,此时用户并未观看人物的头部. 文献[2]和文献[11]的两种分块策略传输了视频的全部内容,如图9(d)所示. 然而,此时VVSTiler选择了一个较细粒度的分块策略(4×12×4),如图9(e)所示,其中人物的头部被划分到一个独立的视频块中,因此没有被传输至客户端. 该结果表明,VVSTiler可以在不造成任何不利影响的情况下



(*a*) longdress第1 184帧时的用户预测视口　(*b*) 文献[2]分块方法客户端的渲染视图　(*c*) VVSTiler分块方法客户端的渲染视图



(*d*) 文献[2,11]分块方法在longdress第1 178帧时客户端的渲染视图　(*e*) VVSTiler分块方法在longdress第1 178帧时客户端的渲染视图

图9　VVSTiler与对比方法的主观定性比较

减少传输数据量,从而减少网络和解码开销.

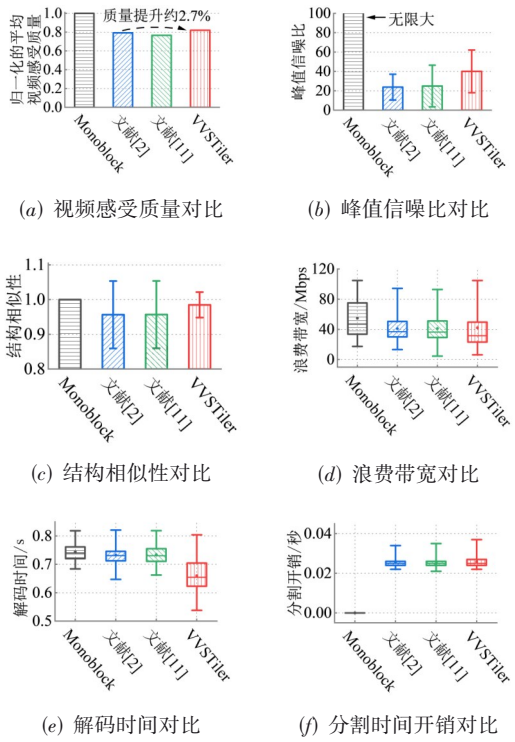
6.7.2　客观定量比较

我们从四个方面定量评估VVSTiler和其他对比方法,包括客观视频质量、网络带宽浪费情况、解码时间和视频分割时间开销. 其中,对于图10(*a*)~(*c*)中的度量指标,数值越大表示视频质量越高;对于图10(*d*)~(*f*)中的度量指标,数值越小表示带宽浪费或时间开销越小.

(1)客观视频质量. 图10(*a*)给出了根据式(5)定义的视频观感质量的对比结果. 其中monoblock由于传输了整个视频,所以得到了最高的质量. VVSTiler比其他两个方法至少提升了2.7%的视频观感质量. 此外,图10(*b*)和图10(*c*)还给出了两个更常用的评价指标上的结果:峰值信噪比(Peak Signal-to-Noise Ratio,PSNR)和结构相似性(Structural SIMilarity,SSIM)[31]. 其中,monoblock对应的PSNR和SSIM值分别为无穷大和1. VVSTiler比其他两个方法在PSNR上平均提高了60.4%,在SSIM上平均提高了2.9%,这得益于当存在视口预测误差时,VVSTiler可通过自适应地选择较粗粒度的分块策略以容忍这种偏差.

(2)网络带宽浪费. 图10(*d*)给出了传输实际视口外的视频内容而导致的带宽浪费情况. 其中monoblock浪费的带宽最多. 相比之下,VVSTiler的平均带宽浪费最少,这是因为当视口预测较准确时,VVSTiler可通过自适应地选择更细粒度的分块策略以减少视口外的点云传输. 具体而言,VVSTiler比表现最好的方法每帧节省了高达27%的带宽.

(3)解码时间. 图10(*e*)比较了视频帧的解码时间. 其中monoblock所需时间最长. VVSTiler比其他方法至少减少了73 ms,这是因为VVSTiler在视口准确时自适应地选择更细粒度的分块策略,避免了需要解码的大量点云;在视口存在偏差时又能选择较粗粒度的分块策略,保证了解码的效率.



(*a*) 视频感受质量对比　(*b*) 峰值信噪比对比　(*c*) 结构相似性对比　(*d*) 浪费带宽对比　(*e*) 解码时间对比　(*f*) 分割时间开销对比

图10　VVSTiler与对比方法的客观定量比较

(4)分割时间. 图10(*f*)给出了分割时间(将整个视频帧分割成小块所需的时间)的对比结果. 越细粒度的分块策略需要越长的时间. 其中monoblock的分割开销为0. VVSTiler由于采用更细粒度的分块策略的情况较多,因此比其他两种方法每帧多0.5 ms左右,这个差距可忽略不计.

7　讨论

VVSTiler是整个全息视频流通信系统里的一个重要组成部分,具有较大的应用价值和优势. 其中一条优势在于,它验证了不同的分块粒度产生的不同影响,这种自适应分块的思想具有广泛的启发作用,特别是对于处理三维体积视频的分块问题. 例如,体积视频的分块不一定非以长方体作为基本几何单位,目前,一些AI驱动的点云压缩传输方法[32]在将点云坐标输入到神经网络之前会把视频帧分割为若干个球形碎片(patch). 然而,这些方法仍采用了固定的分片模式,并传输了整个视频帧. patch的数量过少可能无法充分覆盖整个物体表面,数量过多则会导致patch之间存在大量重叠. VVSTiler的自适应分块思想和求解方案可以为这些方法提供新的思路,帮助它们更好地确定patch数量和形状,以实现更好的AI压缩传输效果.

VVSTiler的另一条优势在于它的独立性,它可以与现有的视频编解码方法直接结合,包括传统的Octree和

kd-tree等编码方式,也包括最近提出的AI驱动的编码方式. 例如,文献[32]通过深度神经网络提取视频的语义特征,然后传输这些特征,最终在接收端进行重建,从而降低数据传输量. VVSTiler可以通过构建基于patch的自适应分块方法与其结合,只提取在用户视口内的视频内容的语义特征,进一步降低传输数据量. 或者,可以直接先使用VVSTiler将视频分块,用AI驱动的方法对需要传输的视频块进行碎片化,特征提取、传输和重建,最后再将这些块组合在一起. 除此之外,VVSTiler的低时间复杂度使得它可以与其他技术结合使用,如点云超分辨率、视频缓存和云渲染技术等,以进一步提高全息视频流的传输和呈现性能. 这种多层次的结合可以帮助满足不同应用场景中对高质量、低带宽和低时延的需求.

8 结论

全息视频的巨大数据量和耗时的编解码计算一直是限制其应用于实时通信场景的主要因素. 本文揭示了现有的分块方法存在的缺陷,深入分析了不同分块策略对视频观感质量和传输效率的显著影响,验证了根据用户视口误差和网络状况动态选择最优分块策略的必要性. 在此基础上,我们专门针对全息体积视频格式设计了一种名为VVSTiler的自适应分块传输方法,以一种简单有效的求解方式避免了复杂问题的优化. VVSTiler在不影响用户体验的情况下减少了传输过程中的数据冗余,保证了编解码的计算效率. 它无需对视频内容进行语义上的理解,适用于各种不同通信场景,且独立于通信过程中的其他组成部分,低计算复杂度的特性便于其与任何编解码器及其他技术结合. 实验结果明确显示,在分割时间开销相近的情况下,VVSTiler在多个方面明显优于其他方法,包括在主观和客观视频质量、带宽浪费以及解码时间等方面,有望显著提升用户体验和系统性能.

参考文献

[1] QIAN F, HAN B, PAIR J, et al. Toward practical volumetric video streaming on commodity smartphones[C]//Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications. New York: ACM, 2019: 135-140.

[2] LIU Z, LI Q Y, CHEN X F, et al. Point cloud video streaming: Challenges and solutions[J]. IEEE Network, 2021, 35 (5): 202-209.

[3] RUSU R B, COUSINS S. 3D is here: Point cloud library (PCL)[C]//2011 IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 2011: 1-4.

[4] GOOGLE. Draco[EB/OL]. (2017-04-15) [2023-07-14]. https://github.com/google/draco.

[5] GRAZIOSI D, NAKAGAMI O, KUMA S, et al. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC)[J]. APSIPA Transactions on Signal and Information Processing, 2020, 9(1): e13.

[6] HOSSEINI M, TIMMERER C. Dynamic adaptive point cloud streaming[C]//Proceedings of the 23rd Packet Video Workshop. New York: ACM, 2018: 25-30.

[7] VAN DER HOOFT J, WAUTERS T, DE TURCK F, et al. Towards 6DoF HTTP adaptive streaming through point cloud compression[C]//Proceedings of the 27th ACM International Conference on Multimedia. New York: ACM, 2019: 2405-2413.

[8] WANG L S, LI C L, DAI W R, et al. QoE-driven adaptive streaming for point clouds[J]. IEEE Transactions on Multimedia, 2022, 25: 2543-2558.

[9] HAN B, LIU Y, QIAN F. ViVo: Visibility-aware mobile volumetric video streaming[C]//Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. New York: ACM, 2020: 1-13.

[10] LEE K, YI J, LEE Y, et al. GROOT: A real-time streaming system of high-fidelity volumetric videos[C]//Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. New York: ACM, 2020: 1-14.

[11] LI J, ZHANG C, LIU Z, et al. Joint communication and computational resource allocation for QoE-driven point cloud video streaming[C]//ICC 2020 - 2020 IEEE International Conference on Communications (ICC). Piscataway: IEEE, 2020: 1-6.

[12] PARK J, CHOU P A, HWANG J N. Volumetric media streaming for augmented reality[C]//2018 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE, 2018: 1-6.

[13] PARK J, CHOU P A, HWANG J N. Rate-utility optimized streaming of volumetric media for augmented reality[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2019, 9(1): 149-162.

[14] SUBRAMANYAM S, VIOLA I, HANJALIC A, et al. User centered adaptive streaming of dynamic point clouds with low complexity tiling[C]//Proceedings of the 28th ACM International Conference on Multimedia. New York: ACM, 2020: 3669-3677.

[15] LI J, ZHANG C, LIU Z, et al. Optimal volumetric video streaming with hybrid saliency based tiling[J]. IEEE Transactions on Multimedia, 2022, 25: 2939-2953.

[16] D´EON E, HARRISON B, MYERS T, et al. 8i voxelized full bodies—A voxelized point cloud dataset[J]. ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) Input Doc-

ument WG11M40059/WG1M74006, 2017, 7(8): 11.
[17] ESRI. Limited error point cloud compression[EB/OL]. (2018)[2023-07-14]. https://github.com/Esri/lepcc/.
[18] QUE Z Z, LU G, XU D. VoxelContext-Net: An Octree based Framework for Point Cloud Compression[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2021: 6038-6047.
[19] HUBO E, MERTENS T, HABER T, et al. The quantized kd-tree: Efficient ray tracing of compressed point clouds [C]//2006 IEEE Symposium on Interactive Ray Tracing. Piscataway: IEEE, 2006: 105-113.
[20] ZHANG L, SUO Y Y, WU X M, et al. TBRA: Tiling and bitrate adaptation for mobile 360-degree video streaming [C]//Proceedings of the 29th ACM International Conference on Multimedia. New York: ACM, 2021: 4007-4015.
[21] XIE L, XU Z M, BAN Y X, et al. 360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming[C]//Proceedings of the 25th ACM international conference on Multimedia. New York: ACM, 2017: 315-323.
[22] YADAV P K, OOI W T. Tile rate allocation for 360-degree tiled adaptive video streaming[C]//Proceedings of the 28th ACM International Conference on Multimedia. New York: ACM, 2020: 3724-3733.
[23] XIAO M B, ZHOU C, LIU Y, et al. OpTile: Toward optimal tiling in 360-degree video streaming[C]//Proceedings of the 25th ACM international conference on Multimedia. New York: ACM, 2017: 708-716.
[24] KELLERER H, PFERSCHY U, PISINGER D. Introduction to NP-completeness of knapsack problems[M]//Knapsack Problems. Berlin: Springer, 2004: 483-493.
[25] POULARAKIS K, IOSIFIDIS G, ARGYRIOU A, et al. Caching and operator cooperation policies for layered video content delivery[C]//IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. Piscataway: IEEE, 2016: 1-9.
[26] ASSARSSON U, MÖLLER T. Optimized view frustum culling algorithms for bounding boxes[J]. Journal of Graphics Tools, 2000, 5(1): 9-22.
[27] Lighthouse3d. View frustum culling[EB/OL].(2017)[2023-07-14]. http://www.lighthouse3d.com/tutorials/view-frustum-culling/index/.
[28] VAN DER HOOFT J, PETRANGELI S, WAUTERS T, et al. HTTP/2-based adaptive streaming of HEVC video over 4G/LTE networks[J]. IEEE Communications Letters, 2016, 20(11): 2177-2180.
[29] HOU X S, ZHANG J Z, BUDAGAVI M, et al. Head and body motion prediction to enable mobile VR experiences with low latency[C]//2019 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE, 2019: 1-7.
[30] JAMALI M, COULOMBE S, VAKILI A, et al. LSTM-based viewpoint prediction for multi-quality tiled video coding in virtual reality streaming[C]//2020 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE, 2020: 1-5.
[31] HORÉ A, ZIOU D. Image quality metrics: PSNR vs. SSIM[C]//2010 20th International Conference on Pattern Recognition. Piscataway: IEEE, 2010: 2366-2369.
[32] HUANG Y K, ZHU Y W, QIAO X Q, et al. Toward holographic video communications: A promising AI-driven solution[J]. IEEE Communications Magazine, 2022, 60 (11): 82-88.

作者简介

朱原玮 男,1997年1月出生于安徽省亳州市. 现为北京邮电大学计算机学院(国家示范性软件学院)、网络与交换技术全国重点实验室博士研究生. 主要研究方向为体积视频、沉浸式视频传输等.

黄亚坤 男,1992年9月出生于安徽省合肥市.现为北京邮电大学计算机学院(国家示范性软件学院)副研究员.主要研究方向为体积视频、移动计算、增强现实等.在国内外期刊及会议上发表学术论文20余篇.

乔秀全 男,1978年6月出生于山西省汾阳市.现为北京邮电大学网络与交换技术全国重点实验室教授,博士生导师.北京邮电大学信息化技术中心副主任(主持工作).北京市科技新星,牵头获中国电子学会技术发明一等奖.主持了6G重点研发计划课题、国家重点基础研究发展计划(973计划)课题等10多项国家纵向项目.主要研究方向为未来互联网、服务计算、计算机视觉、分布式深度学习、增强现实、虚拟现实和5G网络等.在国内外期刊及会议上合作发表学术论文60余篇.中国电子学会会员编号:E190156601M.

E-mail: qiaoxq@bupt.edu.cn