Vues: Practical Mobile Volumetric Video Streaming Through Multiview Transcoding

Yu Liu, Bo Han, Feng Qian, Arvind Narayanan, Zhi-Li Zhang

Abstract

The emerging volumetric videos offer a fully immersive, six degrees of freedom (6DoF) viewing experience, at the cost of extremely high bandwidth demand. In this paper, we design, implement, and evaluate Vues, an edge-assisted transcoding system that delivers high-quality volumetric videos with low bandwidth requirement, low decoding overhead, and high quality of experience (QoE) on mobile devices. Through an IRB-approved user study, we build a first-of-its-kind QoE model to quantify the impact of various factors introduced by transcoding volumetric content into 2D videos. Motivated by the key observations from this user study, Vues employs a novel multiview approach with the overarching goal of boosting QoE. The Vues edge server adaptively transcodes a volumetric video frame into multiple 2D views with the help of a few lightweight machine learning models and strategically balances the extra bandwidth consumption of additional views and the improved QoE, indicated by our QoE model. The client selects the view that optimizes the QoE among the delivered candidates for display. Comprehensive evaluations using a prototype implementation indicate that Vues dramatically outperforms existing approaches. On average, it improves the QoE by 35% (up to 85%), compared to single-view transcoding schemes, and reduces the bandwidth consumption by 95%, compared to the state-of-the-art that directly streams volumetric videos.

Keywords: Volumetric Video Streaming, Multiview Transcoding, Mobile Mixed Reality, Edge Computing, Quality-of-experience (QoE).

摘要

新兴的体积视频提供了完全身临其境的六自由度 (6DoF) 观看体验，但代价是极高的带宽需求。在本文中，我们设计、实现和评估 Vues，这是一种边缘辅助转码系统，可在移动设备上提供高质量的体积视频，带宽要求低，解码开销低，体验质量高 (QoE)。通过 IRB 批准的用户研究，我们构建了首个 QoE 模型，以量化将体积内容转码为 2D 视频时引入的各种因素的影响。受本次用户研究的关键观察结果的启发，Vues 采用了一种新颖的多视图方法，其总体目标是提高 QoE。Vues 边缘服务器借助一些轻量级机器学习模型自适应地将体积视频帧转码为多个 2D 视图，并战略性地平衡额外视图的额外带宽消耗和改进的 QoE（由我们的 QoE 模型表示）。客户端在交付的候选者中选择优化 QoE 的视图进行显示。使用原型实现进行的综合评估表明，Vues 的性能显着优于现有方法。平均而

言，与单视图转码方案相比，它的 QoE 提高了 35%（高达 85%），与直接传输体积视频的最先进技术相比，带宽消耗减少了 95%。

关键词：体积视频流、多视图转码、移动混合现实、边缘计算、体验质量 (QoE)。

1 引言

近年来，随着计算机技术和网络技术的不断进步，视频通信和娱乐行业迎来了一个新的技术突破体积视频（Volumetric Video）。体积视频是一种新兴的视频技术，它可以将真实世界中的三维场景或对象捕捉到一个立体图像中，并实现在不同的角度和位置上观察这个立体图像。与传统的视频技术相比，体积视频可以提供更加真实、自然的观影体验，可以实现用户与视频场景的更加互动式交互。

在为用户带来更真实体验的同时，体积视频也饱受带宽的困扰。以 8i 数据集中的体积视频为例，其中每个视频帧包含大约 1M 个点每个点需要约 15 个字节进行存储，其中 12 个字节用于存储空间坐标，3 个字节用于存储 RGB 属性，以每秒 30FPS 的标准帧速率播放此类视频需要高达 $1 \times 1\,5 \times 3\,0 \times 8 = 3\,.6$Gbps 的带宽。

目前最先进的体积视频流方法可以分为两类：直接流 [18,24,31,41] 和转码流 [16,17,42]。当直接流式传输体积视频时，客户端会在解码和渲染之前下载完整形式或分段部分的编码 3D 内容。尽管它可以提供理想的体验质量 (QoE)，但由于八叉树等复杂的压缩算法，直接流式传输面临着高带宽要求（例如数百 Mbps）和移动设备上不小的解码开销 [25, 38] 和 kd 树 [26, 32]。在转码流系统中，服务器或边缘代理通过基于客户端（预测的）视口将 3D 场景渲染为 2D 图像来执行实时转码，并流式传输编码的 2D 视频。现有的解决方案 [16, 17] 使用简单的预测模型为每个帧创建单个视图。当视口预测不准确时，它们可能会导致 QoE 较差，这对于由于 6DoF 运动动力学而导致的体积视频流很常见。

为了解决上述问题，作者提出了 Vues，这是一种受益于多视图转码的实用移动体积视频流系统。在 Vues 中，边缘服务器从服务器获取原始（高质量）体积内容，然后利用一些轻量级视口预测模型自适应地将每个帧转码为多个 2D 视图。之后，它会流向客户端精心挑选的候选视图，从而提供更多的显示选择，进而提高 QoE。

2 相关工作

2.1 体积视频流

文献中存在一些关于体积视频流的研究 [3–6]。其中，ViVo [5] 引入了几种可见性感知优化，主要用于流式传输体积视频的可见部分，以减少资源消耗。GROOT [31] 是另一个提高点云压缩效率的提案。上述方法直接提供体积视频。最近，Gül 等人 [3, 4] 建议利用云服务器上的远程渲染（即转码）来实现低延迟体积视频流。然而，对于每一帧，他们仅使用单个模型来预测流媒体的单个视口，这可能会由于用户的快速运动降低 QoE。

2.2 基于转码的系统

转码方法已广泛应用于许多其他流媒体系统，包括 360 视频流、云游戏、虚拟现实（VR）等。DeepVista [11] 利用边缘计算将超高分辨率（高达 16K）全景视频流传输到移动设备服务器根据客户端的视口提取和转码视图。Outatime [7] 提出了一种低延迟移动云游戏系统，它将场景渲染卸载到边缘服务器并流式传输转码后的图像。FlashBack [2] 通过广泛的预渲染、转码和存储用户可以离线查看的所有可能图像，实现在移动设备上渲染高质量的 VR 内容。然而以上的系统都没有解决体积视频流带来的独特挑战。

2.3 体积视频的体验质量评估

最近有一些关于直接评估点云视觉质量的工作。例如，Meynet [8] 等人使用数据驱动的方法，提出了彩色点云的全参考视觉质量度量。维奥拉等人 [9] 探索全局颜色统计数据，例如直方图和相关图，以分析基于颜色的指标，以确定点云的损伤程度。Alexiou 等人 [1] 对点云的现有主观质量评估和客观质量指标进行了详细审查。最先进的视觉质量评估侧重于静态点云。

3 本文方法

3.1 本文方法概述

不同于现有的体积视频转码方案只通过单一视口预测向用户提供单一的视口视图，Vues 结合多个视口预测算法，从多个预测的用户视口中进行一定的可能视口扩展，通过一定的冗余应对复杂多变的用户状态行为，并结合了 User-Study 的 QoE 建模对可选视口打分，从而进行发送视口个数的选择和码率自适应的调整。

3.2 QoE 建模

作者通过 User-Study 调研了现有的体积视频影响用户体验质量的关键因素，分别是：

- 视口漂移 D_V，表示预测视口与真实视口的偏移
- 视口平滑度 S_V，指两个连续帧的视口漂移向量之间差异的大小
- 视口移动距离 M_V，指预测的视口在两个连续帧之间的平移距离，其可能比真实视口的移动距离更长
- 运动到光子延迟 MTP，反映用户运动（无论是平移还是旋转）到显示设备上所需的时间

通过结合传统 2D 视频中影响用户体验的因素：分辨率 R，分辨率变化 R_V，卡顿持续时间 B，构建得到用户 QoE 模型如下：

$$QoE = q(D_V, S_V, M_V, L, R, R_v, B)$$

而后通过进一步的用户研究，收集了在不同影响 QoE 因素下用户观影的评分，选取线性回归模型拟合得到的结果得到了最终的 QoE 模型，如图 1。

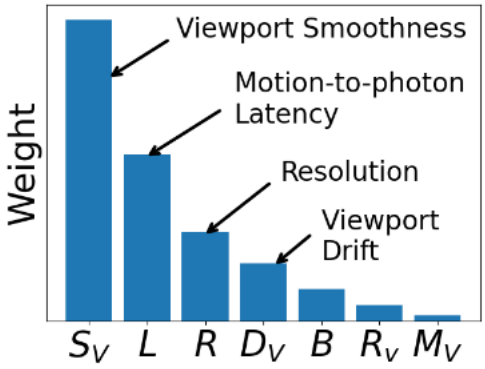

图 1. QoE 因素的重要性

3.3 生成多个候选视图

Vues 利用多个预测（线性回归 LR、多层感知机 MLP、不预测）模型为单个帧创建三个视图。并通过三个预测结果扩展可选的视口视图，如图 2其中蓝色方块是三个模型预测的平移位置，绿色圆圈是 Vues 创建的额外视图的位置。通过将蓝色框的宽度增加 δ_w ，高度增加 δ_h ，以创建外部绿色框。通过选择网格内的 9 个绿色圆圈和外部边界框上的 8 个绿色圆圈来生成额外的候选视图。总共在 Vues 中确定了 20 个候选视图（从扩展区域的 17 个视图加上来自预测模型的 3 个视图）。直观地说，两个相邻网格点之间的距离应该大约等于用户在两个连续帧之间可能移动的步长 Δ_s，以适应视口预测误差。作者通过从 Study-Trace 收集的视口轨迹中推导出步长 Δ_s，大约是 0.1 米。已知 Δ_s，则可以根据预测结果确定的凸包计算 δ_w 和 δ_h：

$$\Delta_s = \frac{1}{4}(2 \times \delta_w + C_w) = \frac{1}{2}(\delta_h + C_h)$$

其中 C_w 和 C_h 分别是凸包的宽度和高度。通过解上述方程，有：

$$\delta_w = 2 \times \Delta_s - \frac{1}{2}C_w, \quad \delta_h = 2 \times \Delta_s - \frac{1}{2}C_h$$

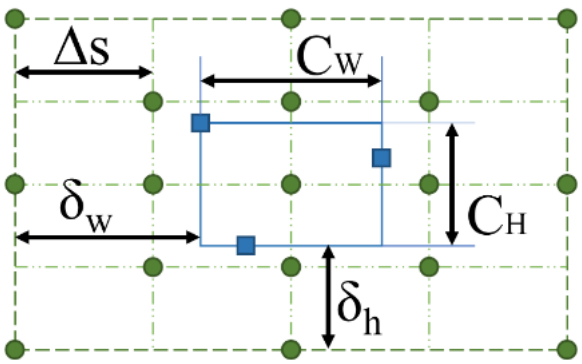

图 2. 选择用于创建额外视图的位置

3.4 整理并传输生成的候选视图

为了将多个候选视图发送给客户端 Vues 提供了两种可选的发送方案。空间排列：将多个视图组合成一个更大的帧（4K）进行传输；时间排列：按顺序对候选视图进行编码，而不将它们合并到单个帧中。

3.5 对生成的候选视图进行打分排序

对候选视图进行打分的关键挑战是估计 QoE 模型中使用的真实视口。尽管每个单独模型的视口预测可能不准确，但通过简单地对这些模型的结果进行平均可以提高预测精度，从而提供对地面实况的良好估计。这种预测方法为 AVG。AVG 方法受到集成学习的启发，它利用多种学习算法来获得比单独使用任何基本学习算法所能实现的更好的预测性能。通过 AVG 估计的真实视口和作者设计得到的 QoE 模型计算便可计算每个候选视图的分数。

3.6 决定视图的个数

对候选视图进行排名后，应该确定将流式传输到客户端的适当视图数量，目标是最大限度地减少带宽消耗，同时不影响用户感知的 QoE。很明显的当用户静止或缓慢移动时，三个预测模型将产生相似的结果。在这种情况下，发送几乎没有差异的多个视图会浪费网络带宽。另一方面，当用户快速移动时，预测不太准确，需要更多视图来补偿预测误差。因此，作者提出了一种启发式算法来根据用户的运动来决定适当的视图数量，而不是固定候选视图的数量。对于每一帧，初始视图数设置为 1。原因是当用户静止时，三个预测模型给出相同的结果，因此不需要发送额外的视图。当用户缓慢移动时（由 $\delta_h = 0$ 和 $\delta_w = 0$ 表示），将为 LR 和 MLP 的预测结果添加两个视图。当用户移动太快或随机时，则为较大的 C_W 或 C_H 添加两个视图，导致每帧最多 7 个视图。

3.7 适应动态的网络

这里不仅需要确定视图的数量，还需要确定每个视图的分辨率；上述两个维度形成了相对于视图数量的指数解空间。为了使解决方案高效，作者开发了一种基于梯度下降概念的近似算法。在该方法中，首先从上述启发式算法确定的视图列表开始，每个视图都具有最高分辨率。然后，迭代地减少视图数量或视图分辨率，直到满足带宽约束。由于视图已按 QoE 分数进行排名，因此在每次迭代中，都采取贪婪操作，删除最后一个（最不重要）视图，或降低高于最低质量的最不重要视图的分辨率。由于所有视图都会产生统计上相似的带宽消耗，因此操纵最不重要的视图（就其 QoE 贡献而言）可确保以最小 QoE 降低为代价来减少带宽。一只重复上述过程，直到剩余视图的带宽使用量小于预测的可用带宽。

3.8 选择播放的视图

Vues 利用提出的 QoE 模型来计算每个候选视图的 QoE 分数，并显示分数最高的视图。为此，Vues 不断跟踪真实视口轨迹和每个显示帧的视口。客户端使用当前帧和先前帧的基本事实以及先前帧的显示视口来计算视口平滑度。上述信息使客户端能够计算每个视图的 QoE 分数，并选择 QoE 最好的一个进行显示。

4 复现细节

4.1 与已有开源代码对比

该篇论文的用户数据集和具体实现代码均未公布，无对应开源代码。工作中查阅资料使用 Unity 对用户观看体积视频数据进行收集。

4.2 实验环境搭建

实验主要是对 Vues 多视图的生成、打分和选取进行实现，并统计最终选取结果计算得到的 QoE、D_V，S_V，M_V 与论文的结果进行对比，验证复现情况，整个实验流程使用 Python 编程实现。

4.3 用户运动轨迹预测

文中采用三种预测方法进行用户运动轨迹的预测，这里根据文中提到了 60% 的情况下用户是静止的进行用户轨迹的采集，并实现了相应的预测方法，具体预测结果如下图 5（左图为预测结果图，右图为预测与真实差的绝对值，即视口漂移的 CDF 图），可见三种预测方法都不能对用户真实的运动轨迹进行良好的预测，更有甚者会与真实的用户运动轨迹有较大的偏差，并且不像真实运动轨迹一般平滑。

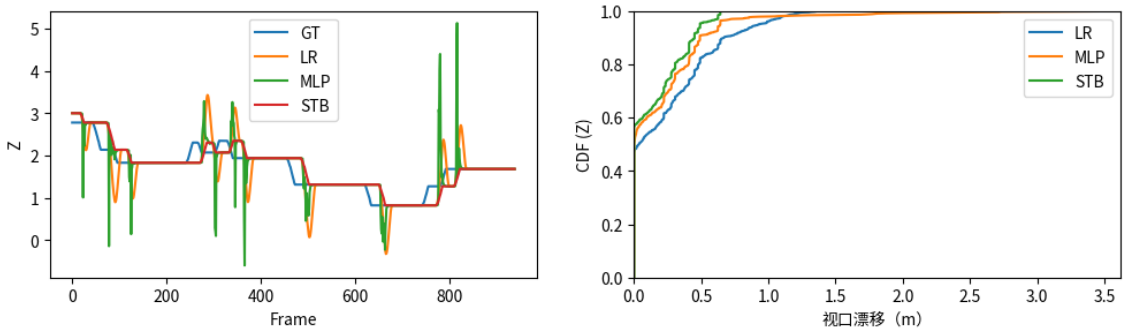

图 3. 用户运动轨迹预测

4.4 多视图的生成

根据 Vues 的设计，其通过获取三个预测方法的预测结果后往外扩充增加可选视图，根据三个预测结果位置首先计算 C_W、C_H 和 δ_w，δ_h，并通过可扩展矩形的一个顶点根据计算得到的参数扩充得到额外的视口视图，具体流程如 Procedure 1。

4.5 多视图的打分排序

通过计算每个候选视图的 QoE 便能对每个视图进行打分并排序（结算 QoE 时只用到了 D_V，S_V 和 M_V），这里 QoE 的参数大小根据作者论文中给出的结果进行设置，具体的处理流程如 Procedure 2。

Procedure 1 多视图的生成

procedure GENERATING MULTIPLE CANDIDATE VIEWS

$x_{array} \leftarrow (\left[\text{pred\_lr\_x[current]} \quad \text{pred\_mlp\_x[current]} \quad \text{pred\_stb\_x[current]}\right])$ ▷ 当前帧的 x 预测

$z_{array} \leftarrow (\left[\text{pred\_lr\_z[current]} \quad \text{pred\_mlp\_z[current]} \quad \text{pred\_stb\_z[current]}\right])$ ▷ 当前帧的 z 预测

$x_{max}, x_{min} \leftarrow \max(x_{array}), \min(x_{array})$ ▷ 计算 x 的最大最小值

$z_{max}, z_{min} \leftarrow \max(z_{array}), \min(z_{array})$ ▷ 计算 z 的最大最小值

$C_W \leftarrow z_{max} - z_{min}$ ▷ 计算 C_W

$C_H \leftarrow x_{max} - x_{min}$ ▷ 计算 C_H

$\delta_w \leftarrow 2 \cdot \Delta_s - \frac{1}{2} \cdot C_W$ ▷ 计算 δ_w

$\delta_h \leftarrow 2 \cdot \Delta_s - \frac{1}{2} \cdot C_H$ ▷ 计算 δ_h

$Ox, Oz \leftarrow x_{min} - \delta_h, z_{min} - \delta_w$ ▷ 计算左上点的 x 和 z 坐标

$psts \leftarrow [...]$ ▷ 根据已计算的参数，生成候选视口视图

end procedure

4.6 最终视图的选择

这里将所有生成的视口视图都提供给客户端进行最终的视口视图选取，选取只涉及到对可选的视口视图进行 QoE 计算，然后选择使得 QoE 最大的视口视图即可。

5 实验结果分析

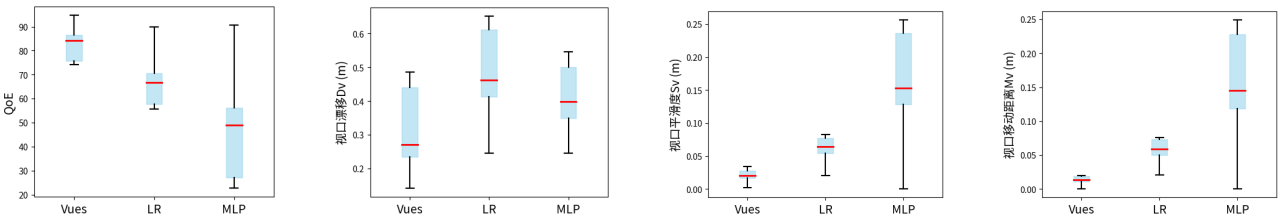

图 4. Vues 和单视图的整体 QoE、视口漂移、平滑度和移动距离（复现）

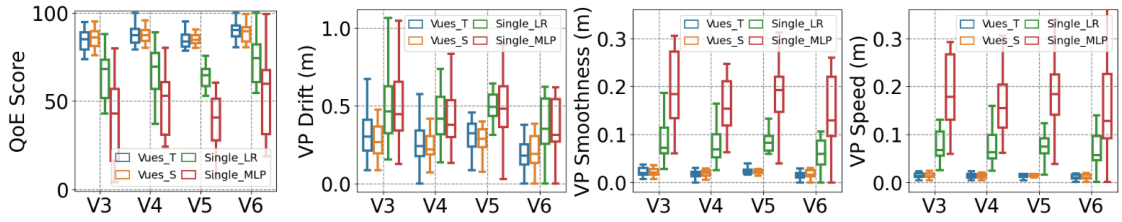

图 5. Vues 和单视图的整体 QoE、视口漂移、平滑度和移动距离（原文）

这里对实验结果的 QoE、D_V，S_V，M_V 与论文的结果进行对比。Vues、LR、MLP 的 QoE 得分中位数分别为 84、67 和 49。与 LR 相比，Vues 的 QoE 提高了 25%，而与 MLP 相比则

Procedure 2 多视图打分排序

procedure RANKING CANDIDATE VIEWS($psts, pst\_gt, previous\_frame, past\_frames, fram\_num$

  $D_V, S_V, M_V, QoE \leftarrow [], [], [], []$ ▷ 初始化当前帧所有视口的视口偏移、平滑度、移动距离和 QoE 列表

  for each pst in $psts$ **do**

    if $frame\_num == 1$ **then** ▷ 初始第一帧

      $D_V^{pst} \leftarrow distances(pst\_gt, pst)$

      $QoE^{pst} \leftarrow 100 - a \cdot D_V^{pst}$

    else if $frame\_num > 1$ **then**

      $D_V^{pst} \leftarrow distances(pst\_gt, pst)$,

      $S_V^{pst} \leftarrow |D_V^{pst} - previous\_frame.D_V| + past\_frames.D_V$

      $M_V^{pst} \leftarrow distance(previous\_frame.pst, pst) + past\_frames.M_V$

      $QoE^{pst} \leftarrow 100 - a \cdot D_V^{pst}/frame\_num - b \cdot S_V^{pst}/(frame\_num - 1) - c \cdot M_V^{pst}/(frame\_num - 1)$

    end if

    $D_V.append(D_V^{pst})$

    $S_V.append(S_V^{pst})$

    $M_V.append(M_V^{pst})$

    $QoE.append(QoE^{pst})$

  end for

  $MAX\_QoE \leftarrow max(QoE)$

  $index \leftarrow QoE.index(MAX\_QoE)$ ▷ 根据最大 QoE 索引存储状态

  $previous\_frame.D_V \leftarrow D_V[index]$

  $previous\_frame.pst \leftarrow pst[index]$

  $past\_frames.D_V \leftarrow past\_frames.D_V + D_V[index]$

  $past\_frames.M_V \leftarrow past\_frames.M_V + M_V[index]$

end procedure

提高了 71%。这里还展示了三个主要因素：视口漂移、视口平滑度和视口移动距离。如图 4所示，Vues 显着提高了视口平滑度，这对整体 QoE 的改善贡献最大。与单一视图相比，它还使得视口移动距离更短。总体而言，结果表明，在多视图的帮助下，Vues 可以通过提高视口平滑度和减少视口移动距离来显着提高 QoE。与论文作者的实验结果（图 5）相符。

6 总结与展望

这项工作中，我对 Vues 设计中影响用户 QoE 的关键指标进行了计算，同时根据 Vues 的思想，完成了整个多视图的生成，打分与选择等，对 Vues 整体进行了完整的实现，实现的效果与论文描述相符。当然这里面我也回避了一些问题，比如 Vues 多视图的时间、空间编码、不同网络带宽环境下 Vues 的性能表现等。

Vues 为体积视频的转码传输提供了基于多视图的解决方案，但其并不能保证用户观看

到的视口视图就是真实的视口视图，其设计旨在从不可能中挑出最优的可能。并且 Vues 的视图选取的性能表现完全依赖于设计实现的 QoE，该 QoE 并没有充足的理论基础而是通过 User-Study，并不能进行大的扩展。

为了让用户能尽可能的收看到真实视口，由于种种限制，光靠服务器显然是不够的，随着硬件的进步发展以及现有的计算机视觉和图像处理工作（NeRF [10] 等）的开展，能否通过利用客户端的硬件资源通过已有的视口逼近真实的视口，在带宽的限制下让用户能得到较好的体验是一个值得考虑的问题。

参考文献

[1] Evangelos Alexiou, Irene Viola, Tomás M Borges, Tiago A Fonseca, Ricardo L De Queiroz, and Touradj Ebrahimi. A comprehensive study of the rate-distortion performance in mpeg point cloud compression. *APSIPA Transactions on Signal and Information Processing*, 8:e27, 2019.

[2] Kevin Boos, David Chu, and Eduardo Cuervo. Flashback: Immersive virtual reality on mobile devices via rendering memoization. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 291–304, 2016.

[3] Serhan Gül, Dimitri Podborski, Thomas Buchholz, Thomas Schierl, and Cornelius Hellge. Low-latency cloud-based volumetric video streaming using head motion prediction. In *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 27–33, 2020.

[4] Serhan Gül, Dimitri Podborski, Jangwoo Son, Gurdeep Singh Bhullar, Thomas Buchholz, Thomas Schierl, and Cornelius Hellge. Cloud rendering-based volumetric video streaming system for mixed reality services. In *Proceedings of the 11th ACM multimedia systems conference*, pages 357–360, 2020.

[5] Bo Han, Yu Liu, and Feng Qian. Vivo: Visibility-aware mobile volumetric video streaming. In *Proceedings of the 26th annual international conference on mobile computing and networking*, pages 1–13, 2020.

[6] Kyungjin Lee, Juheon Yi, Youngki Lee, Sunghyun Choi, and Young Min Kim. Groot: a real-time streaming system of high-fidelity volumetric videos. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[7] Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 151–165, 2015.

[8] Gabriel Meynet, Yana Nehmé, Julie Digne, and Guillaume Lavoué. Pcqm: A full-reference quality metric for colored 3d point clouds. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2020.

[9] Irene Viola, Shishir Subramanyam, and Pablo Cesar. A color-based objective quality metric for point cloud contents. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2020.

[10] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[11] Wenxiao Zhang, Feng Qian, Bo Han, and Pan Hui. Deepvista: 16k panoramic cinema on your mobile device. In *Proceedings of the Web Conference 2021*, pages 2232–2244, 2021.