

基于瓦片的 6-DOF 点云视频的码率自适应算法

唐致烨

摘要

随着虚拟现实和增强现实技术的日益普及，对六自由度（Degree of Freedom, DOF）视频的商业需求也在与日俱增。然而，点云视频通常包括大量的数据，因此很难在带宽有限的网络中进行流式传输。因此，如何在用户的体验质量（Quality of Experience, QoE）能得到高水平保证的情况下，以低带宽消耗为代价，有效地流传资源密集型的点云，是一个极具挑战性的课题。为此，Wang 等人^[1]提出了一种由 QoE 驱动的，基于瓦片的自适应点云视频流媒体传输算法，以最大限度地提高用户的 QoE，同时减少传输的冗余度。通过利用透视投影，他们将每个点云瓦片的 QoE 建模为关于比特率、用户的视域、用户与点云相对位置、点云瓦片间的遮挡关系以及用户终端渲染设备分辨率的函数。基于此，原始 QoE 优化的码率自适应问题得以表述为一个多选择背包问题，即在给定的传输容量下，为不同的点云瓦片分配比特率。这个问题能够被等效地转换为一个受约束的子模函数最大化问题，从而使用贪心算法进行解决。算法能够保证在最坏情况下的理论性能，同时具有非常低的计算时间复杂度。

关键词：点云；码率自适应；透视投影；子模函数最大化；MPEG G-PCC 编解码器

1 引言

三维采集技术的快速发展掀起了一股沉浸式媒体在表现和交互领域的创新浪潮，如虚拟现实（Virtual Reality, VR）、增强现实（Augmented Reality, AR）和混合现实（Mixed Reality, MR）^[2]。这种趋势引起了人们对这些新形式的媒体流的巨大兴趣。其中，360 度视频流的传输在最近几年被广泛研究^[3]。在观看 360 度视频时，用户可以自由调整视角方向三个旋转角度。然而，用户的空间位置是固定的，不能随着用户的运动而改变，这就限制了 360 度视频的应用，即只能支持三个自由度。相比之下，体积数据可以为用户提供六个自由度，允许用户自由改变他们的视角方向和空间位置。常见的体积数据格式是多边形网格（Polygon Meshes）和点云（Point Clouds）。与多边形网格相比，点云表示非流形几何数据的能力更加灵活，更适合于实时处理，不受表面拓扑信息的限制^[2]。因此，基于点云的流媒体形式具有重要意义，因为它能够真正实现用户在沉浸式场景中的自由探索。点云通常由分布在三维空间的点集组成的，每个点都包含一个三维位置坐标 (x, y, z) 和一些其他的属性，如颜色、反射率或表面法线。在沉浸式媒体相关的使用案例中，一个点云通常可能包含 10 万到 1000 万个点的数据，每个分量需要消耗存储空间 8-10 比特^[2]。因此，尽管点云在沉浸式媒体应用中比 360 度视频更有前景，但其更高的自由度需要更大的数据量来支撑，反过来又对目前传输能力有限的网络基础设施造成了更大的负担。由此，当前的主要挑战是如何有效地流传庞大的高质量的点云，从而在保证用户 QoE 的同时保持低的带宽消耗。近年来，人们从两个不同的角度对点云视频的流式传输展开工作。一些工作侧重于将整个点云对象作为速率分配的基本单位^[4-6]，而另一些工作则建议在空间上将点云划分为若干三维瓦片，只传输用户感兴趣的瓦片，以进一步节省传输所需的带宽^[7-8]。这两种路线的共同点是，它们都最大化一些既定的 QoE 指标实现码率的自适应。然而，由于以下两个原因，迄今为止的研究是存在

局限性的。首先，在点云流中，用户的整体 QoE 可能受到许多因素的共同影响^[6,8]，如：1) 瓦片的可用比特率，2) 用户的视域，3) 用户与物体的距离，4) 点云之间的遮挡关系，以及 5) 渲染设备的分辨率。现有的工作都没有考虑到所有这些 QoE 因素。例如，一些工作所设计的 QoE 指标忽略了遮挡和屏幕分辨率的影响，这可能导致带宽的巨大浪费^[6-8]。另一个问题是，这些工作中提出的码率自适应算法都是启发式的，很难在最坏情况下的保证其性能。

为了解决这两个问题，本文提出了一个新的 QoE 指标，将上述所有的 QoE 考虑与计算机图形学中的渲染技术的思想相结合，为点云开发了一种由 QoE 驱动的，基于瓦片的自适应点云视频流媒体传输算法。具体来说，我们将一个点云对象分为若干个三维瓦片，每个瓦片都独立编码为一组比特流，传输到客户端，以避免传输对用户感知的 QoE 贡献不大的不可见瓦片。同时，算法根据各个瓦片对 QoE 的贡献不同，为每个可见瓦片选择适当的质量级别，以便最佳地利用可用带宽。鉴于这些内容通常是渲染在二维的显示设备上，还需要使用透视投影将三维点云投射到二维平面上^[9-10]。

在所提出 QoE 模型下，本文将点云流的最佳码率适应问题表述为一个多选择背包问题 (Multiple-Choice Knapsack Problem, MCKP)，以在带宽约束下最大化用户的 QoE。由于 MCKP 通常是 NP-hard 问题，其精确求解的算法极其耗时，很难满足点云流的低延迟需求。因此，本文将所提出的 MCKP 重新表述为一个等价的子模函数最大化问题，并相应地设计了一个简单而高效的贪心算法，能够以多项式的计算复杂度找到一个接近最优的解决方案。理论证明，所提出的贪心算法的最优近似率下限能够达到 $\frac{1}{2}(1 - 1/e)$ 。

2 相关工作

2.1 点云压缩

点云的压缩技术最近经历了一个快速的发展。为了有效地处理点云数据，当前最先进的点云编解码器一般都使用体素化的方式处理点云数据^[11-12]。体素化的点云，即一组量化为三维立方网格的点 (x, y, z) ，是由体积细分形成的。这种具有边界区域的最小三维立方网格也被称为体素，可以被视为二维图像中像素的三维扩展。如果体素包含原始点云中的任何一个点，就称为被占用。在某一时刻捕获的体素化点云被称为一帧，而动态的体素化点云则由一连串的帧组成。点云压缩 (Point Cloud Compression, PCC) 现在是移动图像专家组 (Moving Picture Experts Group, MPEG) 正在进行的标准化工作的主题之一。从 MPEG 发起的提案征集回应中，有两种压缩技术被选为 PCC 的标准^[13]。1) 基于视频的点云压缩 (Video-based Point Cloud Compression, V-PCC)，三维点云首先被投射到一组二维图像块中，然后被进一步打包成图像。这些图像可以用传统的图像或视频编解码器进行压缩^[14]，使 V-PCC 适合于相对均匀的点云分布。2) 基于几何的点云压缩 (Geometry-based Point Cloud Compression, G-PCC)，它直接在三维空间中对点云进行编码而无需对输入数据坐标进行任何假设，因此适合于更稀疏的点云分布。此外，G-PCC 中引入了用于并行编码的瓦片概念，这与自适应点云流式传输的瓦片方案是一致的。具体来说，在 G-PCC 中，瓦片是具有边界框信息的、可独立编码的区域。因此，本文采用 G-PCC 技术，在流式传输之前对点云进行压缩。

2.2 点云流

近期,有学者探讨了将整个点云对象作为自适应流的速率分配的基本单位的想法^[4-6]。Hosseini 等人^[4]首先提出使用点云下采样的方式产生一个质量较低的点云表示,以节省带宽的想法。对于由多个点云对象组成的场景,他们又开发了一种基于预设参数的启发式码率分配算法^[5]。同时,van der Hooft 等人^[6]提出了三种基于用户注意力和与物体距离的启发式码率自适应算法。具体来说,他们首先根据与用户注意力和用户-点云对象距离相关的一些指标对不同的物体进行排名,然后分别根据三种启发式方法:贪心分配策略、统一分配策略和混合分配策略,将带宽分配给这些不同的对象。然而,正如他们进一步分析所说,这种算法需要根据用户视角的场景和用户的运动轨迹来选择不同的排名指标和分配策略,才能最优化性能^[6],因此并不存在一个适用于所有点云流传输场景的万能方案。

尽管点云压缩技术的效率很高,高质量的点云对象仍需要大量的数据作为支撑。为了更有效地利用传输资源,划分瓦片的方式(最初被广泛应用于 360 度视频流^[15-18])也被引入到点云的码率自适应调度问题中。沿着这一研究方向,已有相当的研究进展。一般的做法是将一个点云分割成若干个三维瓦片,只传输用户视口内的瓦片,以减少传输冗余^[7-8]。具体来说,Li 等人^[7]提出了一个用于指导瓦片选择的 QoE 指标,是基于用户视口、用户和物体间距以及不同瓦片的质量等级设计的。Park 等人^[8]根据瓦片的细节等级(Level of Detail, LOD)为不同的瓦片分配比特率。LOD 被建模为瓦片与用户间距、瓦片与用户视锥关系和显示设备分辨率的函数。

迄今为止的研究都没有考虑到上文中列出的所有可能影响用户 QoE 的因素。这些因素(尤其是物体之间的遮挡)如果被忽略,可能会造成巨大的带宽浪费。此外,现有的速率适应方案都是以启发式方式提出的,缺乏对最坏情况下的性能,即鲁棒性的理论保证。因此,本文的第一个目标是定义一个新的 QoE 指标,它包含了所有 QoE 的考虑,并能明确地描述点云对用户视口的影响。在这个 QoE 指标的指导下,本文的第二个目标是设计一个实际有效的、能在理论上保证其性能稳定性的码率自适应算法。在作者先前的工作^[19]中,他们通过提出一个整体的 QoE 指标来部分解决这些问题,但没有任何理论上的论证;本文则通过利用透视投影来数学地推导出 QoE 指标,并证明了所提出的码率自适应问题可以转换为一个等价的子模态函数最大化问题。

3 本文方法

3.1 QoE 建模

为了在给定的带宽下实现最优化 QoE 的码率自适应,第一个挑战是如何有效地仿真每个点云瓦片对用户 QoE 的贡献。由于最终重建的点云会在显示屏幕上呈现,因此,需要根据用户的视口(即显示屏幕)将每个三维瓦片投影到二维显示平面上,然后测量其效用。由于投影到的是平面而不是曲面,我们主要关注平面几何投影变换^[9],它可以分为两类:平行投影和透视投影。平行投影的投影中心是无限远的,而透视投影的投影中心到投影平面的距离是有限的。因此,在透视投影下,远处的物体看起来比近处的物体要小,这与人类视觉系统的视觉效果是一致的^[9]。事实证明,透视投影比平行投影看起来更真实。基于这种考虑,本文也采用了透视投影。在下文中,将推导出一个透视投影的点云瓦片数学 QoE 模型。



图 1: 利用透视投影将三维坐标转换为二维坐标的流程图

3.1.1 透视投影

首先介绍与透视投影有关的几何变换和矩阵。不同于欧氏几何中常用的笛卡尔坐标，在投影几何领域，主要使用齐次坐标，将投影变换表示成矩阵进行计算。具体来说，齐次坐标的数量要比投影空间多一个维度。例如，一个齐次坐标 (x, y, z, w) 代表欧氏空间中的三维点 $(x/w, y/w, z/w)$ ，其中， w 表示齐次坐标所引入的额外维度。如果 $w=1$ ，则称该点为齐次的。在下面的数学变换中，三维坐标默认是齐次的。在图 1 展示了从三维几何坐标到二维坐标在显示屏幕上的转换流程图。原始三维点云的几何坐标首先与模型-视角矩阵相乘，从物体空间转换到视觉空间，然后通过透视投影将眼睛空间的坐标转换为归一化设备坐标（Normalized Device Coordinates, NDC）。最后，对 NDC 坐标进行视口转换，产生了显示空间的二维坐标。这些变换矩阵的具体介绍如下。

- 模型-视角矩阵。设 $(x_{obj}, y_{obj}, z_{obj}, 1)$ 表示原始三维点云中某一点在物体空间的几何坐标， $(x_{eye}, y_{eye}, z_{eye}, 1)$ 表示视觉空间的坐标，则有

$$\begin{bmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ 1 \end{bmatrix} = M_{modelview} \cdot \begin{bmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ 1 \end{bmatrix} = M_{model} \cdot M_{view} \cdot \begin{bmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ 1 \end{bmatrix}, \quad (1)$$

式 1 中， $M_{modelview}$ 是矩阵 M_{model} 和 M_{view} 的乘积， M_{model} 是将坐标从物体空间转换到世界空间的变换矩阵， M_{view} 是将坐标从世界空间转换到视觉空间的变换矩阵，它们都是平移、缩放、旋转矩阵的组合。

- 透视投影。如图 2 (a) 所示，透视投影中的视锥是一个被截断的金字塔状，它是由垂直于中心轴的一对平行平面（ $z = -n$ 和 $z = -f$ ）截断金字塔而形成的^[20]。视锥中的点首先通过投影矩阵转换为剪除空间中的坐标，然后，通过透视分割将剪除空间坐标映射成 NDC。其中， x 、 y 和 z 坐标的范围都在 $[-1, 1]$ 之内。特别地，设 $(x_{clip}, y_{clip}, z_{clip}, 1)$ 为剪除空间中的坐标，有

$$\begin{bmatrix} x_{clip} \\ y_{clip} \\ z_{clip} \\ w_{clip} \end{bmatrix} = M_{projection} \cdot \begin{bmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ 1 \end{bmatrix}, \quad (2)$$

其中， $M_{projection}$ 表示透视投影矩阵。考虑视锥对称且以 z 轴为中心， $M_{projection}$ 可以被标示为

$$M_{projection} = \begin{bmatrix} \frac{2n}{width} & 0 & 0 & 0 \\ 0 & \frac{2n}{height} & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad (3)$$

$width$ 和 $height$ 表示视锥近平面的宽与高。设用户的垂直视野范围角度为 FoV ，纵横比为 $ratio$ ，则投影矩阵可以进一步表示为

$$M_{projection} = \begin{bmatrix} \frac{1}{ratio} \cot \frac{FoV}{2} & 0 & 0 & 0 \\ 0 & \cot \frac{FoV}{2} & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad (4)$$

更进一步，将剪除空间的坐标转换成 NDC 坐标，使用透视分割

$$\begin{bmatrix} x_{ndc} \\ y_{ndc} \\ z_{ndc} \end{bmatrix} = \begin{bmatrix} x_{clip}/w_{clip} \\ y_{clip}/w_{clip} \\ z_{clip}/w_{clip} \end{bmatrix}, \quad (5)$$

其中， $(x_{ndc}, y_{ndc}, z_{ndc})$ 表示归一化设备空间中点的坐标。

- 视口变换。为了适应二维的渲染显示设备，需要对 NDC 进行视口变换，以生成显示空间的坐标。如图 2 (b) 所示，显示坐标从 $(s_x, s_y, 0)$ 到 $(s_x + w_s, s_y + h_s, 0)$ 不等，屏幕空间中的 z 坐标范围在 $[n_s, f_s]$ 之内。令 $(x_{screen}, y_{screen}, z_{screen})$ 表示显示空间中点的坐标，有

$$\begin{bmatrix} x_{screen} \\ y_{screen} \\ z_{screen} \\ 1 \end{bmatrix} = M_{viewport} \cdot \begin{bmatrix} x_{ndc} \\ y_{ndc} \\ z_{ndc} \\ 1 \end{bmatrix}, \quad (6)$$

其中， $M_{viewport}$ 是视口变换矩阵

$$M_{viewport} = \begin{bmatrix} \frac{w_s}{2} & 0 & 0 & s_x + \frac{w_s}{2} \\ 0 & \frac{h_s}{2} & 0 & s_y + \frac{h_s}{2} \\ 0 & 0 & \frac{f_s - n_s}{2} & \frac{f_s + n_s}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

在视口转换后，显示空间中可能存在多个 z_{screen} 不同的点共享相同的 (x_{screen}, y_{screen}) 。在这些点中，只有 z_{screen} 最小的点 \hat{p}_j 是可见的，因为其他的点都被遮挡了。通过挑选出可见的点集 $\{\hat{p}_j\}$ ，就能得到显示屏幕上呈现的二维坐标，即可见点集 $\{\hat{p}_j\}$ 的 (x_{screen}, y_{screen}) 。

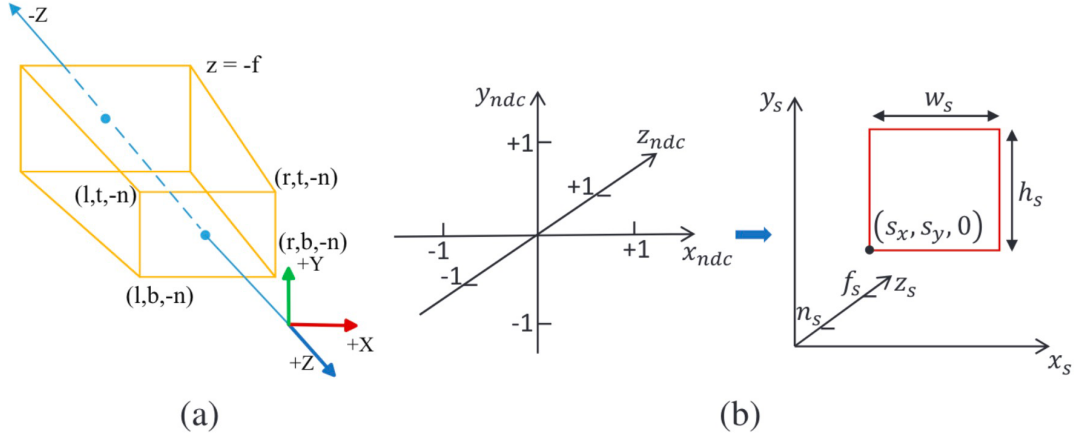


图 2: (a) 通过透视投影产生的视锥。(b) 从 NDC 空间到显示空间的视口变换。

3.1.2 QoE 度量

一个点云瓦片的 QoE 应该由用户的视域、位置、瓦片间遮挡关系、显示设备分辨率以及为该瓦片分配的比特率共同决定。直观地说，如果一个瓦片是不可见的（例如，在用户的视域之外或完全被另一个瓦片遮挡），那么这个瓦片对用户视口的整体视觉质量没有任何贡献。另一方面，对于一个可见的瓦片，它的 QoE 通常被认为应该与瓦片到用户的距离成反比的同时^[7-8]，也受用户显示屏幕分辨率的制约。此外，比特率的影响可以简单理解为，更高的比特率表示对应于更高的视觉质量，从而对用户的 QoE 有更大的贡献。考虑到上述因素，本文将用户视域中质量等级为 q 的三维瓦片 T_i 的 QoE 指标定义为

$$Q_{i,q} = UPB_i - B_{i,q}, \quad (8)$$

其中， $B_{i,q}$ 表示编码瓦片 T_i 的第 q 个质量等级所需的比特数， UPB_i 表示瓦片的每比特效用（Utility Per Bit），由

$$UPB_i = \frac{N_{p,i}}{N_{v,i} \cdot B_{i,max}} \quad (9)$$

所定义。在式 9 中， $N_{v,i}$ 是用户视域中未经压缩的 T_i 所包含的点数量， $N_{p,i}$ 是通过透视投影在显示屏幕上呈现的像素数量， $B_{i,max}$ 是归一化参数，表示以最高质量水平编码 T_i 所需的比特数。从 8、9 可以看出，瓦片与用户视口之间的关系、瓦片与用户之间的距离、不同瓦片之间的遮挡以及屏幕分辨率的约束都通过透视投影隐性地反映在 UPB_i 中，而瓦片的比特率对 QoE 的影响则明确地反映在 $B_{i,q}$ 中。因此，所有可能影响用户视觉质量的关键因素都在这个 QoE 的定义中被考虑。

3.2 最优化 QoE 的码率自适应

3.2.1 优化问题建模

点云流码率自适应的最终目标是最大化用户的 QoE，同时保持尽可能低的带宽消耗。具体来说，需要通过决定每个可见瓦片实际传输的编码质量实现这个最终目标的最优化。因此，QoE 驱动的码率自适应问题可以表述为每个点云帧的 QoE 最大化问题。给定总比特率的预算和当前视锥中可见瓦片的表示集，适当分配相应的比特率来最大化这些可见瓦片的整体 QoE。

为方便表示，设 $T = T_i, \forall i = 1, 2, \dots, N_T$ 为可见瓦片集，每个瓦片编码为不同质量级别的 N_Q 个

表示。对于瓦片 T_i 的第 q 个表示 $T_{i,q}$ ，定义一个二进制变 $x_{i,q}$ 表示 $T_{i,q}$ 是否包含在所选瓦片集中： $x_{i,q}$ 置 1 表示 $T_{i,q}$ 被选中于流式传输，否则置 0。因此，在数学上，这个 QoE 最大化问题可以表述为如下的整数线性规划（Integer Linear Programming, ILP）问题：

$$\begin{aligned}
& \max \quad \sum_{i,q} Q_{i,q} x_{i,q} \\
& \text{s.t.} \quad \sum_{i,q} B_{i,q} x_{i,q} \leq BW \\
& \quad \sum_q x_{i,q} = 1 \\
& \quad x_{i,q} \in \{0, 1\} \quad i \in \{1, \dots, N_T\}, q \in \{1, \dots, N_Q\}
\end{aligned} \tag{10}$$

其中， BW 是总的带宽预算，第二个约束方程强制为每个可见瓦片选择且仅选择一种质量级别的表示。式 10 中的 ILP 问题可以看作是 MCKP^[21] 的一种形式。一般来说，MCKP 具有 NP-hard 的计算复杂度，这点可以通过背包问题（Knapsack Problem, KP）^[21] 进行归约证明。MCKP 可以通过启发式算法^[22-23] 近似解决，也可以通过提供最佳解决方案的方法精确地解决^[24]。尽管启发式算法可以找到复杂度较低的近似解，但通常难以保证最坏情况下的算法性能。相比之下，像分支定界法^[20]、动态编程^[24] 这样确切的方法即使使用商业性优化软件（如 LINGO^[25] 和 CPLEX^[26]）也难以解决大规模的 MCKP。这主要是因为，实现这些确切的方法需要更高的计算复杂度，这也与点云流的低延迟要求相矛盾。因此，另一个挑战是如何具有理论性能保证地快速找到式 10 中 MCKP 的近似解。

3.2.2 等价的子模函数最大化

子模性（Submodularity）是离散域集合函数的一个属性，与连续函数的凹凸性类似。子模函数最大化问题与函数的凹性相关联，因此可以使用简单的贪心算法，在保证性能的情况下获得接近最优的解决方案。为了解决上述问题，需要整理 10 中的 MCKP 为一个受背包约束的、等效的子模函数最大化问题。

特别地，可以从观察式 8 开始讨论。对任意 $1 \leq q_1 < q_2 \leq N_Q$ ，设有 $B_{i,q_1} < B_{i,q_2}$ ，从而有 $Q_{i,q_1} < Q_{i,q_2}$ 。引入两个辅助变量 $\Delta Q_{i,q} = Q_{i,q} - Q_{i,1}$ 、 $\Delta B_{i,q} = B_{i,q} - B_{i,1}$ ，其中 $i \in \{1, \dots, N_T\}$ ， $q \in \{1, \dots, N_Q\}$ ，式 10 中的 MCKP 可以被整理为

$$\begin{aligned}
& \max \quad Q_1 + \sum_{i,q} \Delta Q_{i,q} x_{i,q} \\
& \text{s.t.} \quad \sum_{i,q} \Delta B_{i,q} x_{i,q} \leq BW - B_1 \\
& \quad \sum_q x_{i,q} \leq 1 \\
& \quad x_{i,q} \in \{0, 1\} \quad i \in \{1, \dots, N_T\}, q \in \{2, \dots, N_Q\}
\end{aligned} \tag{11}$$

其中 $Q_1 = \sum_{i=1}^{N_T} Q_{i,1}$ 、 $B_1 = \sum_{i=1}^{N_T} B_{i,1}$ 均为非负常数。式 10 中的等式约束被转换成不等式约束。更进一步，为了将 MCKP 转化成普通背包问题，式 11 可以转化成以下等效形式

$$\begin{aligned}
& \max \quad \sum_i \max_q (\Delta Q_{i,q} x_{i,q}) \\
& \text{s.t.} \quad \sum_{i,q} \Delta B_{i,q} x_{i,q} \leq BW - B_1 \\
& \quad x_{i,q} \in \{0, 1\} \quad i \in \{1, \dots, N_T\}, q \in \{2, \dots, N_Q\}
\end{aligned} \tag{12}$$

删除了式 11 中的常数 Q_1 并重整优化目标函数，我们得到了式 12。主要到约束 $\sum_q x_{i,q} \leq 1$ 也被一并

删除了，这是因为如果存在某解为同一瓦片指定了多个质量等级的表示，较低比特率的点云表示仅消耗带宽却不能为式 1 中的目标函数带来任何正收益，因此不可能是最优解。

3.2.3 算法设计

为了在多项式时间内与理论保证下高效解决式 12 中引入的子模函数最大化问题，本文设计了一个如算法 1 所示的基于贪心策略的算法。在算法 1 中，为简化符号，定义了两个辅助的集合函数 $Q(A) = \sum_{x_{i,q} \in A} \Delta Q_{i,q}$ 和 $B(A) = \sum_{x_{i,q} \in A} \Delta B_{i,q}$ 。该算法返回 $A^* = \bar{A} \oplus \bar{V}$ 作为最优解，其中 \bar{A} 代表基于成本效益（Cost-Benefit）和 QoE 效用（Utility）的贪心搜索得到的近似最优解中的较好的一个， \bar{V} 表示以最低质量等级编码的所有可见瓦片 $T_{i,1}$ 的集合。 \oplus 运算定义如下：

$$\bar{A} \oplus \{x_{i,1}\} = \begin{cases} \bar{A} \cup \{x_{i,1}\}, & \text{if } \nexists x_{i,q} \in \bar{A}, 2 \leq q \leq N_Q \\ \bar{A}, & \text{otherwise} \end{cases} \quad (13)$$

根据^[27]，所提出的码率自适应算法理论上在最坏情况下的性能保证是 $\frac{1}{2}(1 - 1/e)$ 。

Procedure 1 基于贪心策略的码率自适应算法

Input: 基集 $V = \{x_{i,q} | 1 \leq i \leq N_T, 2 \leq q \leq N_Q\}$ ，所有可见点云瓦片在所有质量等级下的 QoE $Q_{i,q}$ 和比特率 $B_{i,q}$ ，带宽预算 BW

Output: 被选中的表示集合 A^*

计算 $\Delta Q_{i,q}$ 、 $\Delta B_{i,q}$ 、 Q_1 和 B_1 , $A_{cb} \leftarrow \emptyset$

while $\exists x \in V \setminus A_{cb} : B(A_{cb} \cup \{x\}) \leq BW - B_1$ **do**

$$x_{cb}^* \leftarrow \underset{\substack{x \in V \setminus A_{cb} \\ B(A_{cb} \cup \{x\}) \leq BW - B_1}}{\operatorname{argmax}} \frac{Q(A_{cb} \cup \{x\}) - Q(A_{cb})}{B(\{x\})}$$

$$A_{cb} \leftarrow A_{cb} \cup \{x_{cb}^*\}$$

end

$A_{ub} \leftarrow \emptyset$

while $\exists x \in V \setminus A_{ub} : B(A_{ub} \cup \{x\}) \leq BW - B_1$ **do**

$$x_{ub}^* \leftarrow \underset{\substack{x \in V \setminus A_{ub} \\ B(A_{ub} \cup \{x\}) \leq BW - B_1}}{\operatorname{argmax}} \frac{Q(A_{ub} \cup \{x\}) - Q(A_{ub})}{B(\{x\})}$$

$$A_{ub} \leftarrow A_{ub} \cup \{x_{ub}^*\}$$

end

$\bar{A} \leftarrow \underset{A \in \{A_{cb}, A_{ub}\}}{\operatorname{argmax}} Q(A)$

$\bar{V} \leftarrow \{x_{i,1} | 1 \leq i \leq N_T\}$

return $A^* \leftarrow \bar{A} \oplus \bar{V}$

4 复现细节

4.1 与已有开源代码对比

本文并未借鉴任何开源代码。

4.2 实验环境搭建

完整的实验仿真环境主要由两个类组成，分别是描述服务器端的 **Server** 类和描述用户端的 **User** 类。图 3 展示了 **Server** 类和 **User** 类的工作流程。服务器端是主模块，负责加载完整的点云数据，并按照体素化的方式对点云内的各个点进行分块（分瓦片）、编码。每个瓦片的质心信息、在每个质量等级下的表示码率将会传输到用户端进行进一步的计算。用户端维护了当前用户的视口与显示设备的相关参数，以将每个瓦片的质心变换到视觉空间，并根据质心与视锥的碰撞与否决定每个瓦片是否进入可见瓦片集 τ 。可见瓦片集的所有点会被融合成一个子点云，供用户端读取。用户端根据图 1 确定可见点集 $\{\hat{p}_j\}$ ，并反索引到每个点所在的原始瓦片，以计算其 UPB_i 函数。

为了方便在原始三维坐标、可见瓦片集、可见点集之间进行频繁的索引与反索引，在实际实现中，为了提升索引性能，我们维护了 $(x_{T_j}, y_{T_j}, z_{T_j}) \leftrightarrow j$ 和 $j \leftrightarrow i$ 两套映射。其中， i 和 j 分别表示在瓦片在全部瓦片中的索引与瓦片在可见瓦片集中的索引。

一旦每个点云瓦片的 QoE 值向量 $Q_i = (Q_{i,1}, Q_{i,2}, \dots, Q_{i,N_Q})$ 被确定，则可以计算出矩阵 $\Delta Q \in \mathbb{R}^{N_T \times N_Q}$ 与 $\Delta B \in \mathbb{R}^{N_T \times N_Q}$ 。用矩阵的方式存储频繁回访的数据，一方面能够提高运算效率，另一方面也能便于调用 NumPy 库所提供的 API 进行高效计算，进一步提升时间性能。

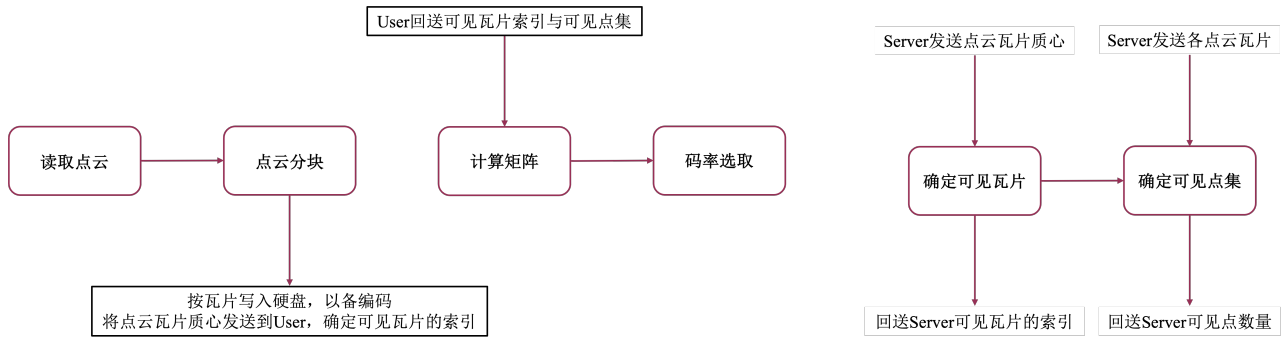


图 3: 服务器端 **Server** 类与用户端 **User** 类的工作流程

5 总结与展望

由于需要在合理的用户视口下实验才能显著可观察的性能，而当前缺乏此类由动态点云与用户视口共同描述的点云数据，因此，实验在点云流调度上的性能并未能进一步实验，需要后续加以实验。同时，本问题只考虑到单用户场景下的码率自适应问题，但实际商业场景下这种限制是严苛的。未来的讨论方向可能是将问题拓展到多用户模式下，使用学习式的算法，结合深度强化学习算法进行解决。

参考文献

- [1] WANG L, LI C, DAI W, et al. QoE-Driven Adaptive Streaming for Point Clouds[J]. IEEE Transactions on Multimedia, 2022.
- [2] SCHWARZ S, PREDA M, BARONCINI V, et al. Emerging MPEG standards for point cloud compression[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2018, 9(1): 133-148.
- [3] ZOU J, LI C, LIU C, et al. Probabilistic tile visibility-based server-side rate adaptation for adaptive 360-degree video streaming[J]. IEEE Journal of Selected Topics in Signal Processing, 2019, 14(1): 161-176.

- [4] HOSSEINI M, TIMMERER C. Dynamic adaptive point cloud streaming[C]//Proceedings of the 23rd Packet Video Workshop. 2018: 25-30.
- [5] HOSSEINI M. Adaptive rate allocation for view-aware point-cloud streaming[J]. arXiv preprint arXiv:1911.00812, 2019.
- [6] 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. 2019: 2405-2413.
- [7] 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). 2020: 1-6.
- [8] 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.
- [9] FOLEY J D, VAN DAM A, FEINER S K, et al. Introduction to computer graphics: vol. 55[M]. Addison-Wesley Reading, 1994.
- [10] BAKER M, et al. Computer graphics, c version[J]., 1997.
- [11] De OLIVEIRA RENTE P, BRITES C, ASCENSO J, et al. Graph-based static 3D point clouds geometry coding[J]. IEEE Transactions on Multimedia, 2018, 21(2): 284-299.
- [12] LIU Q, YUAN H, HOU J, et al. Model-based joint bit allocation between geometry and color for video-based 3D point cloud compression[J]. IEEE Transactions on Multimedia, 2020, 23: 3278-3291.
- [13] 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.
- [14] JANG E S, PREDA M, MAMMOU K, et al. Video-based point-cloud-compression standard in mpeg: From evidence collection to committee draft [standards in a nutshell][J]. IEEE Signal Processing Magazine, 2019, 36(3): 118-123.
- [15] ALFACE P R, MACQ J F, VERZIJP N. Interactive omnidirectional video delivery: A bandwidth-effective approach[J]. Bell Labs Technical Journal, 2012, 16(4): 135-147.
- [16] YU M, LAKSHMAN H, GIROD B. Content adaptive representations of omnidirectional videos for cinematic virtual reality[C]//Proceedings of the 3rd International Workshop on Immersive Media Experiences. 2015: 1-6.
- [17] LIU X, XIAO Q, GOPALAKRISHNAN V, et al. 360 innovations for panoramic video streaming[C]//Proceedings of the 16th ACM Workshop on Hot Topics in Networks. 2017: 50-56.
- [18] SUN L, DUANMU F, LIU Y, et al. Multi-path multi-tier 360-degree video streaming in 5G networks [C]//Proceedings of the 9th ACM Multimedia Systems Conference. 2018: 162-173.

- [19] WANG L, LI C, DAI W, et al. Qoe-driven and tile-based adaptive streaming for point clouds[C] // ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2021: 1930-1934.
- [20] WANG Z, JIA K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection[C] // 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2019: 1742-1749.
- [21] KELLERER H, PFERSCHY U, PISINGER D. The multiple-choice knapsack problem[G] // Knapsack Problems. Springer, 2004: 317-347.
- [22] AKBAR M M, RAHMAN M S, KAYKOBAD M, et al. Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls[J]. Computers & operations research, 2006, 33(5): 1259-1273.
- [23] HIFI M, MICHRAFY M, SBIHI A. Heuristic algorithms for the multiple-choice multidimensional knapsack problem[J]. Journal of the Operational Research Society, 2004, 55(12): 1323-1332.
- [24] DUDZIŃSKI K, WALUKIEWICZ S. Exact methods for the knapsack problem and its generalizations [J]. European Journal of Operational Research, 1987, 28(1): 3-21.
- [25] SYSTEM L. LINGO 18[Z]. <https://www.lindo.com>.
- [26] IBM. ILOG CPLEX Optimization Studio[Z]. <https://is.gd/3GGOFp>.
- [27] LESKOVEC J, KRAUSE A, GUESTIN C, et al. Cost-effective outbreak detection in networks[C] // Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007: 420-429.