一种用于视口相关360度视频流传输的最优可伸缩视频编码比特流架构

Gang Shen*NEXIntel*美国希尔斯伯勒 gang.shen @intel.com

Mingyang Ma中国科学技术大学软件工程学院中国合肥 mamingyang@mail.utstc.edu.cn

Guangxin Xu*DCAIIntel*中国上海 g ua ngxin.xu@intel.com

arXiv:2304.05654v1 [cs.MM] 12 Apr 2023

摘要——**摘要.ał 为了通过互联网传输超高分辨率的360度视频（例如8K、12K甚至更高分辨率），视口相关流传输对于节省带宽已成为必要手段。在视口切换期间，客户端和服务器将针对给定视口即时交换协调信息和内容。然而，这些视口切换给视频编码带来了严峻挑战，因为变化视口内内容之间的时间依赖是不可预测的。在现有实践中，人们普遍注意到，比特流中的图像组（GOP）大小从根本上限制了视口切换延迟的降低，例如从运动到光子（MTP）延迟或从运动到高质量（MTHQ）延迟。本文提出了一种基于可伸缩视频编码（SVC）的比特流架构，该架构能够在结构上消除图像组对视口相关流传输的影响，并在一帧时间内实现即时视口切换（这是可能达到的最佳效果）。此外，结合分块技术，这种新的编码方案能够高效打包360度视频视口内的非相邻区域。我们的实验还表明，与多流方案相比，采用这种基于SVC的方法进行整体编码的速度更快。与当前基于MPEG-I OMAF的360度视频流解决方案相比，我们的方法在视口切换延迟、视口打包的简便性和编码性能方面更具优势。**

索引词—**360度视频、OMAF、可伸缩视频编码（SVC）、动作到光子（MTP）、虚拟现实（VR）、AV1**

I. 背景

超高分辨率（如 8K、12K 甚至更高）的 360 度视频内容在市场上越来越受欢迎。与此同时，即便采用高效视频编码或 AV1 等现代编解码器，支持此类高分辨率内容所需的带宽仍是网络面临的挑战。据估计，若使用高效视频编码并采用中等质量设置进行编码，支持一路 8K 60 帧率视频流 [1] 可能需要 80–100 Mbps。

球面上的大部分内容在观看过程中对客户端设备不可见，因此，视口相关流传输——即仅传输客户端视场（FOV）内的内容——直观上成为一种可行的（也是唯一可行的）节省带宽和实现大规模分发的方法。在 MPEG-I OMAF（全向媒体格式）中，规定了两种配置文件：基于高级视频编码（AVC）的视口相关视频配置文件和基于高效视频编码（HEVC）的视口相关视频配置文件；此外，还规定了三种相应的基于瓦片的流传输方法 [2], [3]。为了支持视口相关流传输，OMAF 在其中一种方法中专门采用 HEVC MCTS（运动约束瓦片集），将 360度视频帧分割为可独立解码的瓦片，从而能够对任意视口中的内容进行封装、组合和传输。不同平台（包括 Android、iOS、Windows 和 Web）上均有多种商业实现 [4]–[7]。此外，VR 视频设计人员可以受益于基于诺基亚云流处理平台构建的 OMAF 创作系统；最终用户可通过该系统上传、编辑和预览 6K 360度视频，并通过视频链接观看 [8]。虽然最新的基于 OMAF v2 的端到端系统能够处理并显示 8K、60 帧率的 VR 视频 [9], ，但仍面临计算量大、文件封装复杂以及视口切换延迟等挑战。本文提出一种基于可伸缩视频编码（SVC）的 360度视频最优码流方案，以应对 OMAF 当前面临的挑战。尤其是，该方案为视口切换期间的交互延迟提供了卓越的解决方案，而这对于视口相关流传输至关重要且十分常见。下文中，第二节阐述 360度视频格式和流传输面临的挑战；第三节介绍这种基于 SVC 的新编码方案的设计；第四节给出基于 AV1 SVC 的参考实现；第五节展示编码实验和性能分析；第六节总结全文。

II. 视口相关的 360度视频流面临的挑战

A. 视口变化与 *MTP/MTHQ* 延迟

视口相关流传输的成功取决于视口信息与覆盖该视口的部分内容之间交换的效率。尽管这种交换（如图 1 所示）能够节省大量带宽，但会在服务器与客户端之间引入交互延迟，从而可能在视口变化期间损害视觉质量。此外，为支持大规模分发，图 1 中的服务器无法切实依赖实时转码，为众多客户端各自动态的视口准备视频流。因此，亟需一种最优的内容（比特流）模式，以支持超低延迟交换和可扩展分发。

当使用单一内容质量时，MPEG-I将交换/交互延迟定义为 MTP（运动到光子）延迟；当使用两个级别的内容分辨率/质量时，则定义为 MTHQ（运动到高质量）延迟。人眼对检测视觉质量差异十分敏感——根据 [10], ，要为用户带来完全沉浸式的体验，这一延迟需要低至 50 毫秒，而如今市场上的 VR 或 360度视频产品几乎无法做到这一点。

业界已有一些方法可通过视频编码、打包或网络传输来降低这种延迟。目前，MPEG-I OMAF 中流行的方法是结合多码率或多分辨率打包、逐区域打包（RWP）、比特流重写和基于瓦片的编码，以实现节省带宽的方案 [2]。不过，这种方法只能通过增加采用长、短图像组（GOP）尺寸的额外编码轨道来缓解这种交互延迟——也就是说，在视口变化期间，内容将从短 GOP 轨道中选取并传输。但这种方法会增加转码工作量，也会增加传输短 GOP 轨道所需的带宽。

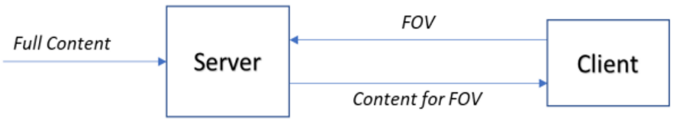


图 1. 视口相关流传输中的视口与内容交换

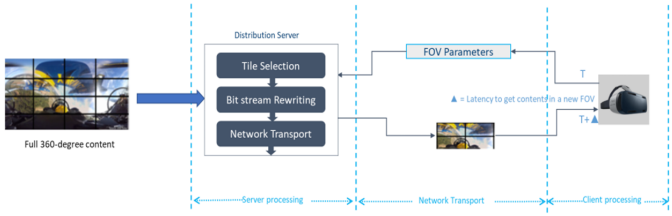


图 2. 视口相关的 360 度视频流传输分发服务器实现示例

当实时VR/360 视频视口相关流传输中发生视口变化时，对内容的访问会同时发生空间上的变化——从一个位置转移到另一个位置，以及时间上的变化——从当前帧转移到下一帧。分发服务器（图2）通常位于边缘网络，将从源端接收完整内容，并根据客户端发送的视场，在比特流中选择部分内容。然而，只有在 IDR帧——图像组的第一帧——处，部分内容的选择才能有意义地进行。图像组结构越长，实现视口变化所需的延迟就越长；与此同时，图像组越短，码率就越高。因此，在视口相关的 360 视频实时流传输的商业实现中，GOP大小是一个难以确定的因素。

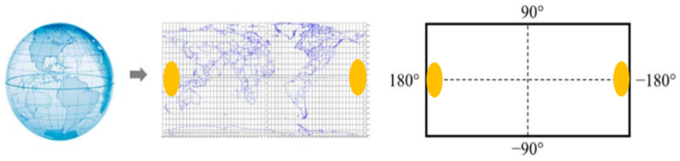


图 3. MPEG-I OMAF 中的等距柱状投影与打包（ [3]）

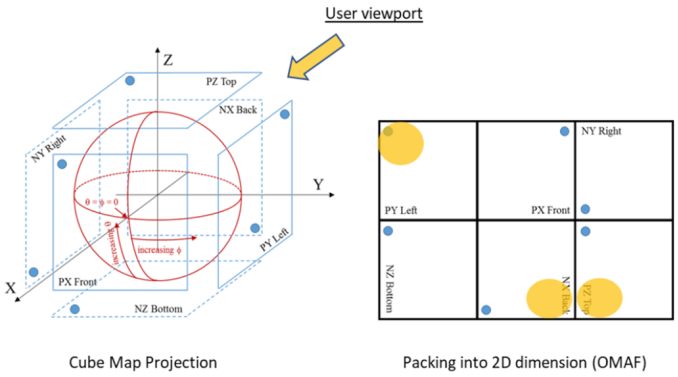


图 4. MPEG-I OMAF ([3]) 中的立方体贴图投影与打包

B. 视口内容的投影与打包

与此同时，360度视频需要采用特定的投影和打包方法，将其编码为二维矩形帧，以便进行编码和网络传输。由于内容从球形空间投影到平面空间，给定视场中的内容在二维帧中并不总是相邻的。这会给视场内容的收集、排列和打包带来困难和复杂性，而通常编码器或解码器的接口都要求将这些内容放入一个矩形区域中。

例如，在等距矩形投影中（如图 3 所示），视场可能由左右边缘处的非相邻区域（以橙色椭圆标出）组成。

立方体贴图投影存在同样的问题。如图4所示，假设：(1)给定的视口来自右上后角；(2) 内容球体被投影到一个立方体上；(3) 立方体的六个表面根据OMAF规范被打包成一个二维矩形区域 [3], 该视口的内容分布在以橙色椭圆标记的非相邻区域中。

给定视口中的这些非相邻区域，对逐区域打包（RWP）、服务器端基于瓦片的编码，以及客户端设备中解码器的延迟绑定都构成了挑战。即使采用基于瓦片的视频编码（如高效视频编码），仍需要精细的码流重写算法，该算法在 MPEG-I OMAF v2.0（ [3], 4.2.3）中有明确规定，用于将这些非相邻区域打包成二维矩形区域，或生成供客户端设备上的一个或多个解码器使用的符合规范的码流。将多个区域打包进一个矩形区域在几何上十分困难，而且使用填充也会带来高昂成本。

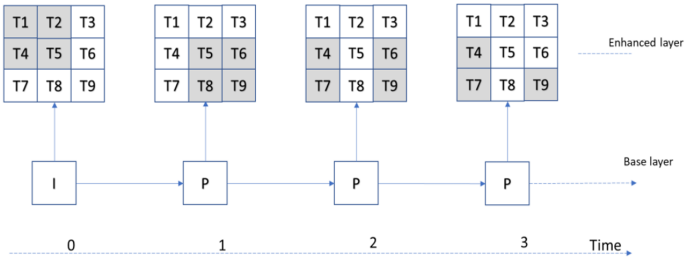


图5. 用于视口相关360度视频流的最优编码方案

III. 一种最优的基于可伸缩视频编码的比特流方案

根据第 II 节的讨论，视口相关的 360度视频流在视频编码的时间域方面面临挑战——GOP大小对交互延迟的影响；在视频编码的空间域方面也面临挑战——给定视场中的非相邻区域。这表明，360度视频可能需要一种先进的编码方案。

事实上，传统视频编码同样面临适应用户端各种需求并提升编码性能的挑战，因此分别发展出了可伸缩视频编码和图块。可伸缩视频编码的主要概念是将视频流拆分为基础层和增强层，从而提供可部分移除的视频流。图块将视频帧划分为相互独立的部分，每个图块都是由超块组成的矩形，其空间参考范围限制在图块边界之内。同一帧中的不同图块可以分别编码，从而支持多线程。

因此，我们提出了一种基于可伸缩视频编码和图块的最优视频编码方案（图 5），该方案将克服 MPEG-IOMAF 中传统编码所面临的困难。

这一新编码方案中有两个关键要素：(1)在可伸缩视频编码中使用增强层传输高质量内容，同时使用基础层传输低质量内容；(2) 使用图块（图块之间没有编码依赖）将一帧分割成多个区域。

在 MPEG-I OMAF 中，基于 AVC 和基于 HEVC 的视口相关 OMAF视频配置通过多路流，将视场中的高质量内容与低质量内容的完整视图相结合（图 6）。在客户端设备上进行渲染时，当高质量内容无法覆盖视场时，低质量内容将作为备份。类似地，SVC 的基础层可以作为备份，而增强层的瓦片可以为任意给定的视场组成内容（甚至是非相邻区域）。

此外，增强层只能与基础层存在空间依赖。这意味着该结构中的时间依赖仅存在于基础层。由于基础层中的所有帧都会传输到客户端，因此增强层中的高质量瓦片可以不受图像组限制而自由访问。

编码方案的详细设计如下：

1）它包含两层：作为基础层的低分辨率（或低质量）层，以及作为增强层的高分辨率（或高质量）层。还可以扩展为多个增强层（例如，用于同步广播）。此外，基础层可以具有时间依赖，即可以包含 I 帧、P 帧和图像组（GOP），从而获得视频压缩方面的优势。本提案中的 GOP 指的是“封闭GOP”。

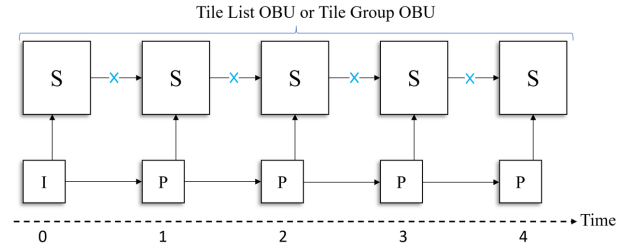
2) 增强层中的帧将仅对基础层具有空间依赖。在图5 中，增强层与基础层之间的依赖关系位于同一帧，但增强层中的帧也可以对基础层中前面的若干帧具有这种依赖关系：例如，增强层中的第 3 帧可能依赖于基础层中的第 1 帧或第 2 帧。关键是，要避免增强层中的帧之间产生时间依赖。

3) 基础层中的帧会依赖多个先前的基础层帧。例如，基础层中的帧 3 可能依赖基础层中的帧 1 或帧 2。4) 基础层可能仅包含单个瓦片/切片，以节省码率。5) 向客户端设备进行流式传输时，只会传输增强层中指定视场范围内的内容（灰色瓦片）以及基础层中的完整瓦片。每个瓦片的位置都会记录在每帧比特流中的清单消息中（例如，高效视频编码中的 SEI 消息 [11] 或 AV1 中的 OBU 标头）。

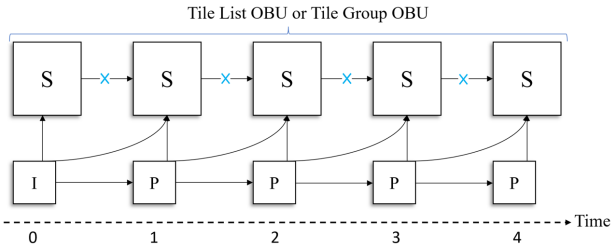
在 360 度视频（或全景视频）流传输过程中发生视口变化时，这种编码结构可以即时、随机地访问增强层中的任意瓦片。如果比特流能够容纳每个瓦片的位置信息，就可以解决视口中不相邻区域的问题，从而无需采用 MPEG-I OMAF 中定义的额外打包方法。

图 5 中的灰色瓦片分别表示时间 0、1、2、3 时给定视场对应的内容/瓦片。当然，视场不一定会在相邻帧之间发生如此剧烈的变化。此处旨在说明该码流方案的可行性和灵活性：

1）从帧 0 到帧 1，新的视场需要解码新的 T6、T8、T9 瓦片（T5 已存在于帧 0 中；T5 可解码）。由于这些瓦片仅依赖基础层，因此可以解码。2）帧 2 显示，新视场的内容包含彼此不相邻的瓦片：T4、T6、T7、T9。如果 SEI 消息（或头部信息）能够提供每个瓦片的位置信息，这仍然是可行的。3）帧 3 显示，这种结构允许包含数量不同的瓦片。这对某些格式（例如 ERP）的360度视频很有用，因为投影可能会导致视口中的像素分布不均。



(a) 增强层指一个帧



(b) 增强层指多个帧

图 6. AV1 可伸缩视频编码中的 L2T1 结构及增强

因此，通过将内容编织成瓦片和层，这种编码方案消除了图像组结构的限制，也免去了区域打包的需要。此外，瓦片和层是 AV1 和 VVC 等现代编解码器内置的概念或工具。基于这种编码方案，可以大幅简化 360度视频流解决方案的端到端实现。

IV. 使用 AV1 可伸缩视频编码的参考实现

这种结构可通过现代编解码标准实现。例如，在 AV1 可伸缩视频编码中，可基于 L2T1 [12]，并添加额外的瓦片规范来构建该结构。

图 6 展示了 AV1 可伸缩视频编码中的 L2T1 可伸缩结构。为支持本提案中的结构，需要进行一些扩展：

1) 移除增强层中帧与帧之间的时间依赖（在图 6 中以 a˘rX.a´s 标记）。2) 基础层可编码为单个瓦片，以节省码率。增强层需要编码为多个瓦片，以便快速切换视点。3) 使用“Tile Group OBU”或“Tile List OBU”选择并打包给定视场的图块。根据当前 AV1 可伸缩视频编码规范，对于这种情况，“Tile Group OBU”更为合理。4) 每个图块都应独立打包或定界——这有助于在传输层独立处理高分辨率图块。5) 增强层可以像图 6(a) 那样参考基础层中的一帧，也可以像图 6(b) 那样参考基础层中的多帧。

这里展示了一个参考实现，说明如何使用“Tile Group OBU”来支持视口切换并传输视场内容，如图 7 所示：

1) 编码器将为整幅图像编码一个高质量层，并通过分块将完整内容划分为多个小区域。根据 AV1 可伸缩视频编码，可在这些图块上应用多个 Tile Group OBU；在这种情况下，每个 Tile Group OBU 只覆盖一个图块——图 7 中的“T#”指一个图块。基础层采用相同的分块方式，但分辨率或质量较低；即使基础层只有一个图块，也不会影响该算法。编码器的输出如图 7 中的 (1) 所示。

2) 服务器将接收编码器 (1) 输出的完整内容；同时，服务器还将接收来自客户端的视口信息（视场信息）——图 7 中仅显示了一个客户端。

3) 服务器将根据特定客户端提供的视口信息，对完整内容帧（1）执行瓦片选择和重新打包（比特流重写）流程。瓦片选择将根据视口参数（图 8），映射为观众视场内的瓦片列表——例如图 7 中（2）所示的 T1、T2、T4 和 T5。然后，服务器将提取这些瓦片（依据 AV1 可伸缩视频编码的语法），并将它们重新打包成一个新的“帧”，其中增强层仅包含视场内的瓦片，而基础层包含所有瓦片。

a) 由于每个瓦片都位于一个 Tile Group OBU 中，且该 Tile Group OBU 不包含其他瓦片，因此可通过 OBU 中的“tg_start”和“tg_end”值确定瓦片的位置。
b) 重新打包所选瓦片后，应添加一个时间分隔符 OBU，用于标识帧与帧之间的边界。c) 此步骤的输出——视场内容，如图 7 中（2）所示。视场外的瓦片将被跳过。

4)一旦客户端接收到所有高分辨率瓦片组（灰色瓦片），它就可以从基础层生成未选中和未接收的图块作为“跳过”图块（蓝色图块，见图 7 的 (3)），前提是：

a) 在帧级别，将累积分布函数（CDF）更新设为禁用，并将全局运动矢量（MV）设为零。b) 在瓦片级别，将变换跳过参数设为 true，并将“使用全局 MV”设为 true。5) 根据 AV1 SVC 规范，生成的这些“跳过”图块可解码。解码后，增强层中跳过图块的区域将通过提取基础层中的对应区域并将其放大来填充。无需对解码器进行任何更改。随后，可以实现一个比特流重写器来选择并

c相应地构造一个“视口相关帧”（如图 中的（2）所示）。它将重写
7每个增强层中的每个目标瓦片。
la层。目标瓦片中的每个超块都将被重写为
th相同的结构。根据 AV1 规范，
sy 超块语法应重写如下：1）**划分**语法写为 None。2）当没有变换系数时，**跳过** 语法写为 True。

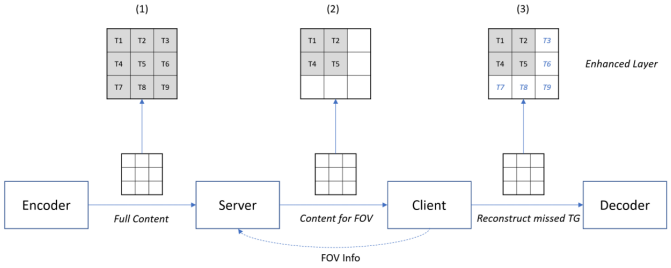


图 7. 所提出的端到端比特流处理



图 8。AV1 比特流 OBU 结构重写器概览

3）**is_帧间** 语法为 True，且仅指基础层。4）运动矢量设为零。5）根据规范，**use_obmc** 语法设为 False，以使用简单平移。

以下是用于比特流重写器的伪代码，用于创建图 7 中的“视口相关帧”(2)：

算法 1： 比特流重写器

```
foreach 增强层 do
    对于每个 目标瓦片执行
        对每个 超块 执行
            重写_sb(超块)
```

算法 2： 重写_sb

写入_分区_模式(*PARTITION_NONE*)$_2$ 写入_跳过(*True*)$_3$ 写入_是_帧间(*True*)$_4$ 写入_参考_帧(*REF_TO_BASE_LAYER_ONLY*)$_5$ 写入_帧间_模式(*零_mv*)$_6$ 写入_使用_*obmc*(*False*)

图 9 展示了 GOP 大小如何影响 MTP/MTHQ 延迟。整个视口切换过程包含多个不同因素，例如姿态检测、网络传输、缓冲、解码和渲染。其中，GOP 大小是一项结构性因素，它会阻止在非 I 帧上立即选择视口相关内容。如果没有像本文所述的分层比特流结构，服务器就必须等待下一个 I 帧，才能响应视口变化，平均耗时约为 GOP 大小的一半。例如，如果 GOP 大小为 10，帧率（每秒帧数）为 30，那么 GOP 大小导致的延迟将为 167 毫秒。尽管这种延迟

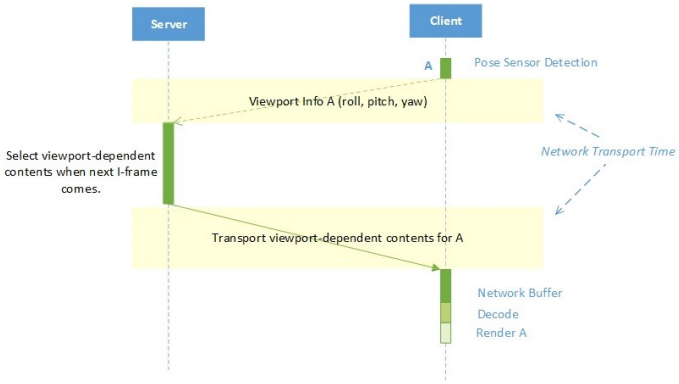


图 9. AV1 比特流 OBU 结构重写器概览

可通过使用较短GOP大小的独立轨道来减少（例如使用GOP大小为3或5），但该轨道会大幅增加码率和编码复杂度。

V. 实验

A. 环境设置

我们扩展了开放媒体联盟（AOM）的 AV1 编解码器，以实现该编码方案，并与现有的 360度视频方法进行了比较。虽然新的编码方案能够实现即时视口变化，并且比现有方法先进得多（如第 3.1 节所示），但我们希望从以下两个方面进一步评估这一新的编码方案：（1）相同码率下的质量；（2）编码时间/速度。

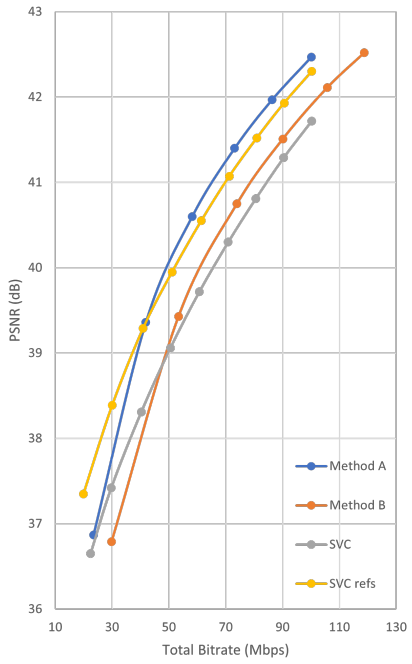
使用 IntelXeonE5-8280 CPU 和192GB内存处理8K和 4K 视频编码。实验使用了三个 30帧率的8K 和4K 360度视频片段（见表 I）；ERP截图如图 12 所示。编码参数设置如下：*CPU*使用量=*7*，遍数=*1*，线程数=*20*，使用终止方式=*CBR* 和实时 模式。每种情况将测试三次，并取所有实验结果的平均值。

表 I 视频片段信息

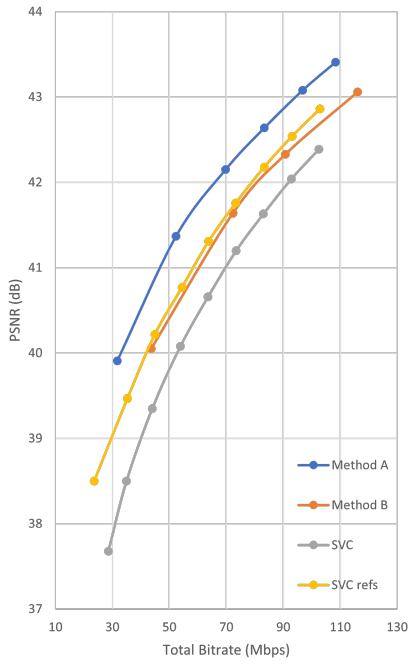
Name	描述	长度	来源	分辨率
东京	东京街道上的人们	23s	[13]	7680x3840
办公室	办公室里的人们	41s	[13]	7680x3840
海滩	俯瞰加勒比海海滩	60s	[14]	7680x3840
Lion	野外的狮子	30s	[15]	3840x1920
Ski	滑雪记录	30s	[16]	3840x1920
街道	街景	34s	[13]	3840x2400

B. 码率与质量

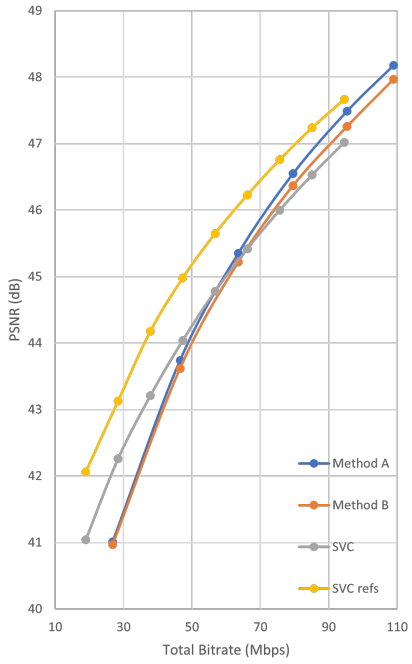
在业界，360度视频解决方案通常包含两到三路视频流，例如，一路用于长时间观看的长图像组大小高分辨率轨道、一路作为背景的长图像组大小低分辨率轨道，以及一路用于视口切换的短图像组大小高分辨率轨道。为使评估更有意义，我们比较了两种方法在选定



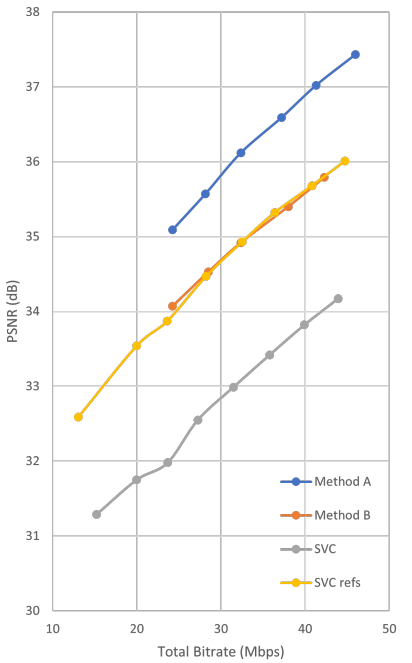
(a) Toyko



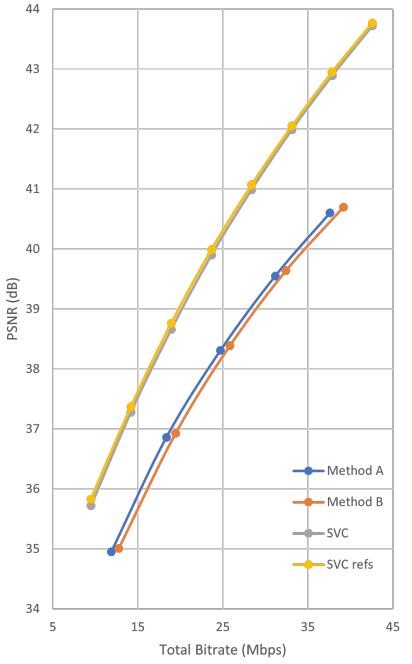
(b) 办公室



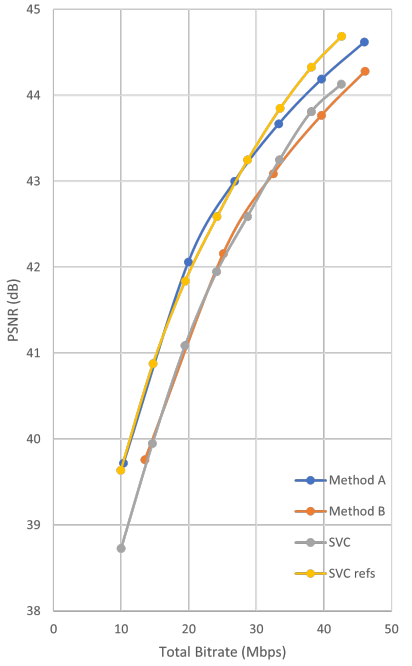
(c) 海滩



(d) 狮子



(e) 滑雪



(f) 街道

[{"id": 0, "output": "图10。在不同编码配置下的峰值信噪比"}]

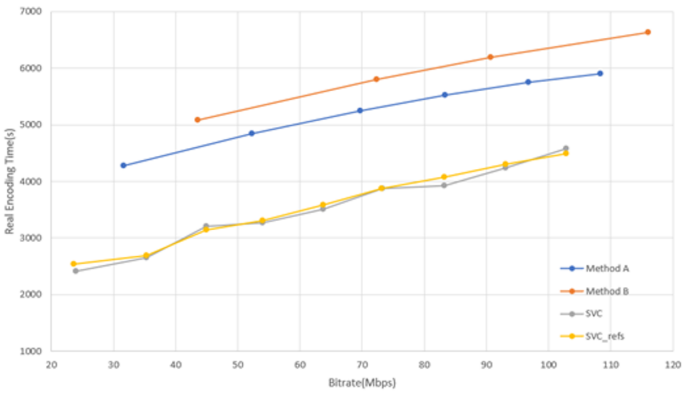


图11. 不同配置下的编码时间

视频片段。对于现有方案，峰值信噪比是两条高分辨率轨道的平均值，可伸缩比为 2.0x，这意味着低质量视频的宽度和高度均为原始视频的一半。

在图10和图11中，方法A和方法B是上述常见的三轨方法，长图像组的大小为30，短图像组的大小分别为5和3。新的基于SVC的编码方案包含图6中的两种结构：增强层仅对应基础层中的一帧或多帧（在图10和图11中分别记为“SVC”和“SVC参考帧”）。

图10显示，基于SVC的方法和三轨方法能够实现相近的单位码率质量。与其他方案相比，我们的方法在特定场景或视频内容中的表现可能略好或略差。此外，值得注意的是，由于图6(b)中的SVC结构，峰值信噪比可提高约0.5–1dB。

另一方面，传输码率也是我们应该考虑的因素之一。根据前文讨论，我们的方案可将延迟降低至一帧的时间。如果现有方法也希望将延迟降低至一帧，则需要传输完整画面，或者至少根据指定场景（如 [17]）提供足够的余量。此处我们不讨论针对特定视频场景的传输策略，因为我们的方法只需要约六分之一的增强层，并相应简化了策略调整，这意味着可显著节省比特流。

C. 编码时间

我们比较了各方法在不同编码码率下的编码时间。结果通过 Linux 系统中的 time 命令记录，等于用户时间与系统时间之和。

图11展示了基于SVC的方案与现有三轨方法的性能对比。结果表明，在相同码率下，新型基于SVC的编码方案的实际编码时间显著缩短（>30%），这意味着在多码率（或多分辨率）应用场景中，采用SVC可以大幅降低编码成本。

VI. 结论

对于360度视频（以及虚拟现实内容），视口相关流传输已成为必要。但现代视频编解码器采用的时间依赖性使视口切换延迟与图像组大小相关，因此难以实现即时视口变化，而这对用户体验至关重要。本文提出了一种基于SVC和瓦片的视频编码结构，以应对视口切换挑战，并将交互延迟缩短至一帧时间——这是理论上可能达到的最短时间。其次，通过在SVC层中使用瓦片，该结构避开了非相邻区域的打包问题；这是非平面内容（如360度视频或虚拟现实内容）中常见的几何挑战。此外，与现有行业方法相比，这种新型编码方案在实现相同码率下相近峰值信噪比的同时，可大幅节省编码成本。

另外，AV1 空间可伸缩视频编码的编码性能可能是一个值得进一步研究的课题。在我们的实验中，基础层与增强层之间采用 2 倍的空间缩放因子，因为可伸缩视频编码通常需要较小的缩放因子才能获得更好的质量 [18]。根据视频内容的不同，AV1 可伸缩视频编码与传统单层编码相比，可能呈现不同的质量指标；但在我们的案例中，峰值信噪比差距可以接受——即使在最差情况下，峰值信噪比差距也小于 1 dB（通常为 0.5 dB）。

最终，这种新编码方案可通过 AV1 SVC [12]，实现，无需更改现有编解码标准，也就是说，无需更改 AV1 解码器。类似的方法也可在 MPEG-I VVC 中通过子图和参考图像重采样（RPR） [19] 实现。此外，采用这种基于可伸缩视频编码的编码方案，可以大幅简化 MPEG-I OMAF [3] 规定的 360 度视频传输架构，例如应用、编码编解码器和封装格式。这项工作为实现最高效的 360 度视频视口相关流传输铺平了道路，涵盖交互延迟、编码性能和架构等方面。

参考文献

[1] Atsuro Ichigaya 和 Yukihiro Nishida。使用高效视频编码/H. 265 的新型广播服务所需比特率分析。*IEEE* 广播学报，62(2):417–425，2016。[2] Miska M Hannuksela、Ye-Kui Wang 和 Ari Hourunranta。360 视频 OMAF 标准概述。载于 *2019* 年数据压缩会议 *(DCC)*，第 418–427 页。IEEE，2019。[3] B Choi、YK Wang、MM Hannuksela、Y Lim 和 A Murtaza。信息技术——沉浸式媒体编码表示（MPEG-I）——第 2 部分 2，2017 23090：全向媒体格式。*ISO/IEC*，页码 – 。[4] O. V. Cloud。沉浸式视频示例。[5] Tiledmedia。ClearVR 如何驱动并利用标准。[6] Tiledmedia。Tiledmedia 瓦片流媒体支持 8K 360VR 流媒体播放。[7] Dimitri Podborski、Jangwoo Son、Gurdeep Singh Bhullar、Robert Skupin、Yago Sanchez、Cornelius Hellge 和 Thomas Schierl。MPEG 360 VR 瓦片流媒体的 HTML5 MSE 播放：采用高效视频编码图块的 MPEG-OMAF 视口相关视频配置文件的 JavaScript 实现。载于 第 *10* 届 *ACM* 多媒体系统会议论文集，第 324–327 页，2019。[8] Yu You、Ari Hourunranta 和 Emre B Aksu。OMAF4Cloud：由标准支持的 360° 视频即服务创作。*SMPTE* 动态影像学报，129(9):18–23，2020。

[9] QiZhang、Jianchao Wei、ShansheWang、Siwei Ma 和 WenGao。Re-alVR：基于 MPEG OMAF 的高效、经济且以体验质量为驱动的 VR 视频系统。*IEEE Transactions on Multimedia*，2022。[10] Rachel Albert、Anjul Patney、DavidLuebke 和 JoohwanKim。虚拟现实中注视点渲染的延迟要求。*ACMTransactions on Applied Perception (TAP)*，14(4):1–13，2017。[11] Gary J Sullivan、Jens-Rainer Ohm、Woo-Jin Han 和 Thomas Wiegand。高效视频编码（高效视频编码）规范概述。*IEEETransactions oncircuits and systems for video technology*，22(12):1649–1668，2012。[12] PeterDeRivaz和 Jack Haughton。AV1 比特流与解码过程规范。*The Alliance for Open Media*，681，2018。[13] Insta360。Insta360 Pro8K 示例。[14] Airpiano。8K中的加勒比天堂 360 视频–YouTube。[15] N. Geographic。狮子 360°。[16] Airpiano。RosaKhutor 滑雪资源。[17] Mehmet N Akcay、Burak Kara、Saba Ahsan、AliC Begen、Igor Curcio和Emre Aksu。用于改善沉浸式视频用户体验的头部运动感知视口边缘。在 *ACM Multimedia Asia*，第 1–5 页。2021。[18] Mathias Wien、Heiko Schwarz 和 Tobias Oelbaum。可伸缩视频编码的性能分析。*IEEE Transactions on Circuits and Systems for Video Technology*，17(9):1194–1203，2007。[19] Benjamin Bross、Ye-Kui Wang、Yan Ye、Shan Liu、Jianle Chen、Gary J Sullivan 和 Jens-Rainer Ohm。多功能视频编码（VVC）规范及其应用概述。*IEEE Transactionson Circuits and Systems for Video Technology*，31(10):3736–3764，2021。



(a) Toyko (b) 办公室 (c) 海滩

(d) 狮子 (e) 滑雪 (f) 街道

图 12. 不同编码配置下的峰值信噪比