

三维投影变换辅助的云游戏编码优化算法

王一帆¹, 王昊¹, 孙浩楠¹, 尘原¹, 霍俊彦¹, 王豪², 张伟^{1,3}

(1. 西安电子科技大学通信工程学院, 710071, 西安; 2. 阿里巴巴集团, 310000, 杭州;

3. 鹏城实验室人工智能研究中心, 518055, 广东深圳)

摘要: 针对传统视频压缩技术无法对云游戏视频中复杂的全局相机旋转以及剧烈变化的大范围运动进行很好的运动估计与补偿的问题,提出了一种三维投影变换辅助的编码优化算法,以提高云游戏视频的编码效率。分析了三维游戏场景下一点投影到二维平面的详细空间变换过程,在此基础上通过建立约束进行最优化估计构建了虚拟相机的内参估计算法。利用估计的内参,结合相机运动信息,基于图像单应性理论生成了与待编码帧平行的辅助参考帧,利用其更新参考帧列表参与编码。由于生成的辅助参考帧与待编码帧之间只有平移运动,所以更有利于传统的运动估计和补偿。实验结果表明:与原始的高效视频编码(HEVC)相比,所提算法在云游戏视频上可以实现平均 8.14% 的 BD-rate 增益,一定程度上缓解了云游戏对网络带宽的高需求;特别地,在仅有相机旋转的情况下,BD-rate 可达 -24.68%,同时编码时间平均缩短 5.34%;与 HEVC-SCC 相比,所提算法的 BD-rate 性能增益超过 20%。

关键词: 云游戏编码;三维投影变换;内参估计;参考帧生成

中图分类号: TN911.7 **文献标志码:** A

DOI: 10.7652/xjtub202312003 **文章编号:** 0253-987X(2023)12-0020-10

Cloud Gaming Video Coding Optimization Algorithm Assisted by Three-Dimensional Projection Transformation

WANG Yifan¹, WANG Hao¹, SUN Haonan¹, CHEN Yuan¹,

HUO Junyan¹, WANG Hao², ZHANG Wei^{1,3}

(1. School of Telecommunications Engineering, Xidian University, Xi'an 710071, China; 2. Alibaba Group, Hangzhou,

310000, China; 3. Artificial Intelligence Research Center, Peng Cheng Laboratory, Shenzhen, Guangdong 518055, China)

Abstract: For the problems that traditional video compression techniques are poor in motion estimation and compensation for complex global camera rotation and abrupt large-scale motion in cloud gaming videos, a coding optimization algorithm assisted by the three-dimensional projection transformation was proposed to improve the coding efficiency for cloud gaming videos. Firstly, the detailed spatial transformation process of projecting a point from a three-dimensional gaming scene to a two-dimensional surface was analyzed. Then, an intrinsic parameter estimation algorithm for the virtual cameras was constructed base on the analysis by establishing constraints for optimal estimation. With the estimated intrinsic parameters and the camera motion information, an auxiliary reference frame parallel to the frame to be coded was generated based on image homography theory and used to update the reference frame list for coding. As the generated auxiliary reference frame

收稿日期: 2023-05-18。 作者简介: 王一帆(1995—),女,博士生;霍俊彦(通信作者),女,副教授。 基金项目: 国家自然科学基金资助项目(62101409,62171353)。

网络出版时间: 2023-07-06

网络出版地址: <https://kns.cnki.net/kcms2/detail/61.1069.T.20230706.1041.002.html>

only involves translational motion with the frame to be coded, it is more conducive to traditional motion estimation and compensation. The experimental results show that, compared with the original HEVC, the proposed algorithm can achieve 8.14% BD-rate gains for cloud gaming videos on average, relieving the high demand for network bandwidth of cloud gaming to some extent. In particular, the BD-rate can be up to -24.68% in the case of only camera rotation. At the same time, the average coding time is reduced by 5.34%. In addition, compared with HEVC-SCC, the proposed algorithm can also achieve the BD-rate gains of more than 20%.

Keywords: cloud gaming video coding; three-dimensional projection transformation; intrinsic parameter estimation; reference frame generation

作为一种新兴的以云计算为基础的游戏方式,云游戏^[1]引起了人们的极大关注^[2]。该场景下,云端服务器完成游戏的复杂运算与渲染,并将 3D 游戏场景压缩为 2D 视频流,然后通过高速宽带网络传输到玩家游戏终端。终端只需要基本的视频解压能力和可靠的网络即可享受到良好的游戏体验。与传统游戏模式相比,云游戏摆脱了终端对硬件的依赖,在很大程度上减少了玩游戏的设备成本。一些工作^[3]提高了云游戏视频传输的可靠性和稳定性,但是高帧率以及高分辨率的游戏视频也对传输带宽提出了更高的要求^[4],例如分辨率为 720P、帧率为 60 帧/s 的游戏视频需要 15 Mb/s 的传输带宽,而对于 4K 视频则需要 40 Mb/s 的传输带宽才可满足用户的质量需求^[5]。传输数据量大、占用带宽高等问题是限制云游戏进步和发展的重要因素。因此,如何提高云游戏视频的编码码率、减小带宽,是目前云游戏亟待解决的一大问题。

一种直观的方式是采用传统的视频压缩技术,如高效视频编码(H.265/HEVC)^[6]等,通过重复利用视频间的相同像素信息来去除冗余。自 20 世纪 80 年代以来,传统视频编码技术蓬勃发展并持续不断优化:文献 [7-9]通过优化编码过程中的滤波、预测等模块来提高编码效率;文献 [10-11]提出了快速模式判决算法来降低编码复杂度;文献 [12-13]建立了更加准确的率失真模型以对码率控制进行改进。此外,随着新兴视频的发展,文献 [14-15]还总结了面向屏幕视频内容的专门编码工具。然而,传统的视频压缩技术及其改进虽然对运动缓慢单一的平移运动具有良好的作用,却无法对相机旋转发挥很好的估计和补偿作用^[16]。在云游戏,尤其是第一人称云游戏中,玩家的视角变化非常频繁,游戏视频往往包含更复杂的旋转动作。此外,游戏视频相邻帧之间存在剧烈变化的大范围运动,传统的亚像素精度运动估计也无法精准地捕捉到相邻帧之间的变化。因此,

直接采用传统的视频编解码技术压缩游戏视频编码,残差较大,码率较高,仅通过调整帧率、分辨率和编码码率等参数^[17-19],难以实现有效的编码优化。

近年来,一些工作基于感兴趣区域(region of interest, ROI)对云游戏视频的编码开展了码率控制。文献 [20]采用显著性图和优先级图合成游戏注意力地图,并利用其设置编码参数,实现高质量的关注区域和较低质量的非关注区域以适应码率限制;文献 [21]针对游戏视频提出了基于多目标优化的编码树单元级别的感知比特分配方法,优化了感兴趣区域的质量同时兼顾了质量波动。结合云游戏视频的内容特性,根据特定的准则将编码比特向重要性高的感兴趣区域倾斜,可以大幅度提高云游戏视频的感知质量。

与由真实摄像机捕获的传统视频序列不同,云游戏视频是计算机生成视频,可以直接从游戏引擎中获得对象信息等图形渲染上下文辅助编码。文献 [22]将来自游戏引擎的部分信息馈送到视频编码器中以绕过运动估计过程,降低计算复杂度,从而提高视频编码速度;文献 [23]利用游戏开发者为游戏对象分配的简单标签提取 ROI 信息,在视频帧中的不同区域分配不同数量的比特来优化视频编码,提高视频的感知质量。此类方法由于需要游戏引擎信息,往往只适用于少量开源游戏平台。此外,由于云游戏序列是计算机根据客户端对虚拟相机的控制指令在服务器端渲染而成的,可以额外获得游戏序列中逐帧的相机运动信息。一些工作^[16,24-26]尝试利用这一特点,通过基于深度图的图像绘制技术(depth image based rendering, DIBR)^[27],将已知图像映射到其他待编码帧处,对映射后图像与原始图像的差值图像进行编码传输。相比于传统的运动估计和运动补偿,DIBR 作为一种基于图像的渲染技术,可以利用逐像素深度以及变换矩阵更为准确地抓取复杂运动间的变换信息,降低码率。然而,此类方法需

要额外的比特去传输深度信息,这在有限带宽的高传输压力下往往是不可接受的。同时,该方法需要虚拟相机的内参进行二维成像平面的投影,由于商业化电子游戏的版权保护,内参无法被直接获得,这使得利用三维投影变换对云游戏编码优化受到了开源云游戏平台限制。这些都会增加优化方案的实施难度。并且,上述工作仅对传统编码的输入进行预处理,并未对编码过程进行实质性地优化。文献[28]利用相机旋转信息,采用深度学习的方法生成增强参考帧,并将其添加到参考帧列表以对现有视频编解码器进行优化。深度学习的使用在提升性能的同时也在一定程度上增加了编码复杂度,且该工作仅针对相机旋转下的游戏视频进行了评估,具有一定的局限性。

基于此,利用云游戏视频易于获得全局相机运动信息这一特点,本文提出了一种三维投影变换辅助的云游戏编码优化算法。该算法首先对游戏场景中世界坐标系下点投影到二维平面的过程进行详细分析,并据此三维投影变换过程建立约束,对相机内参进行估计。接着,利用估计的内参以及已知的摄像机运动信息,利用图像单应性生成与当前待编码帧平行的辅助参考帧,将其添加到编码器的参考帧列表中参与编码。由于生成的辅助参考帧与待编码帧之间只有平移运动,所以更有利于传统的运动估计和补偿。本文算法不需要已知的相机内参与深度信息,使利用三维投影变换对云游戏编码优化变得一般化,同时引入了额外的辅助参考帧参与编码,以编码块为单位降低编码残差,大幅度提高了压缩效率。考虑到 H. 265/HEVC 为目前云游戏应用中的主流视频编码方式,本文将所提优化算法应用到 HEVC 中并与原始 HEVC 进行比较。实验结果表明,本文算法在云游戏视频上平均可以实现 8.14% 的 BD-rate 增益,在仅有相机旋转的情况下,BD-rate 可达 -24.68%。同时,还在一定程度上降低了编码复杂度,编码时间平均缩短 5.34%。此外,本文算法相比于 HEVC 屏幕内容编码扩展(HEVC screen content coding, HEVC-SCC)也有超过 20% 的 BD-rate 性能增益。

1 游戏场景中的三维投影变换

游戏场景中的一点要经过多个坐标空间变换才能实现从三维空间坐标到二维像素坐标的映射。以 Unity 游戏引擎为例,假设三维场景中一点的世界坐标为 $\mathbf{P}_{\text{world}} = (x_w, y_w, z_w, 1)$, 其具体的三维投影变换的过程如图 1 所示。

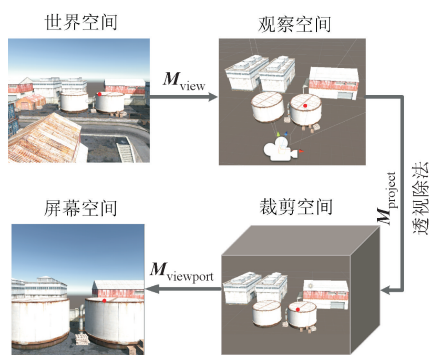


图 1 游戏中的三维投影变换

Fig. 1 Three-dimensional projection transformation in cloud gaming

由于玩家观察场景时总是从摄像机的视角出发,摄像机决定了引擎渲染游戏的画面,因此首先需要将世界空间下的点 $\mathbf{P}_{\text{world}}$ 通过观察矩阵 $\mathbf{M}_{\text{view}}$ 变换到观察空间中

$$\mathbf{P}_{\text{view}} = \mathbf{M}_{\text{view}} \mathbf{P}_{\text{world}} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{T}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (1)$$

式中: $\mathbf{R}$ 和 $\mathbf{T}$ 分别表示从世界坐标系到相机坐标系的旋转和平移变换。

将该点从观察空间变换到裁剪空间,以对相机的可见范围进行设置。裁剪空间由 6 个裁剪平面包围的视锥体决定,其中近裁剪平面和远裁剪平面决定了摄像机可以看到的深度范围。为投影到二维平面进行准备,通过投影矩阵 $\mathbf{M}_{\text{project}}$ 得到顶点在裁剪空间中的坐标 $\mathbf{P}_{\text{clip}} = (x_c, y_c, z_c, w_c)$

$$\mathbf{P}_{\text{clip}} = \mathbf{M}_{\text{project}} \mathbf{P}_{\text{view}} = \begin{bmatrix} \frac{\cot(V'/2)}{r} & 0 & 0 & 0 \\ 0 & \cot \frac{V'}{2} & 0 & 0 \\ 0 & 0 & -\frac{F+N}{F-N} & -\frac{2NF}{F-N} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} = \begin{bmatrix} x_v \frac{\cot(V'/2)}{r} \\ y_v \cot \frac{V'}{2} \\ -z_v \frac{F+N}{F-N} - \frac{2NF}{F-N} \\ -z_v \end{bmatrix} \quad (2)$$

式中: V' 、 r 、 N 、 F 分别表示相机的视场角、近裁剪平面的宽高比、视锥体近裁剪平面和远裁剪平面距离

相机的距离; $\cot$ 是余切函数。由式(2)可知,裁剪坐标 w_c 与 z_v 密切相关,且随着像素的变化而变化。

完成所有裁剪工作后,需要把视锥体投影到屏幕空间来生成对应的 2D 坐标。首先,通过

$$\mathbf{P}_{\text{NDC}} = \frac{1}{w_c} \mathbf{P}_{\text{clip}} = \left(\frac{x_c}{w_c}, \frac{y_c}{w_c}, \frac{z_c}{w_c}, 1 \right) \quad (3)$$

利用透视除法将视锥体变换到一个立方体内,得到归一化的设备坐标 $\mathbf{P}_{\text{NDC}} = (x_{\text{NDC}}, y_{\text{NDC}}, z_{\text{NDC}}, 1)$, 其中 $x_{\text{NDC}}, y_{\text{NDC}}, z_{\text{NDC}} \in [-1, 1]$ 。接着,利用

$$\begin{bmatrix} u \\ v \\ z_{\text{NDC}} \\ 1 \end{bmatrix} = \mathbf{M}_{\text{viewport}} \mathbf{P}_{\text{NDC}} = \begin{bmatrix} \frac{W}{2} & 0 & 0 & \frac{W}{2} \\ 0 & \frac{H}{2} & 0 & \frac{H}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\text{NDC}} \\ y_{\text{NDC}} \\ z_{\text{NDC}} \\ 1 \end{bmatrix} \quad (4)$$

通过简单的视口变换 $\mathbf{M}_{\text{viewport}}$, 根据图像的宽度 W 和高度 H 得到对应的像素坐标 $\mathbf{P}_{\text{pixel}} = (u, v, 1)$ 。

通过上述步骤实现了从摄像机视角出发,将场景从三维空间映射到二维画面的过程。那么,从屏幕上一投影到世界空间的过程则为上述三维投影变换的逆过程。因此,在已知深度信息以及相机内外参(内参即投影矩阵,外参即观察矩阵)的情况下,世界坐标系下一点从一幅视图映射到另一幅视图的变换过程为式(5)

$$\mathbf{M}_{\text{view}} = \mathbf{R}_z \mathbf{R}_x \mathbf{R}_y = \begin{bmatrix} \cos\theta_z \cos\theta_y - \sin\theta_z \sin\theta_x \sin\theta_y & -\sin\theta_z \cos\theta_x & \cos\theta_z \sin\theta_y + \sin\theta_z \sin\theta_x \sin\theta_y & 0 \\ \sin\theta_z \cos\theta_y + \cos\theta_z \sin\theta_x \sin\theta_y & \cos\theta_z \cos\theta_x & \sin\theta_z \sin\theta_y - \cos\theta_z \sin\theta_x \sin\theta_y & 0 \\ -\cos\theta_x \sin\theta_y & \sin\theta_x & \cos\theta_x \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

式中: $\theta_x, \theta_y, \theta_z$ 分别表示绕 x, y, z 轴的旋转角。

已知帧缓存中相对旋转的两幅视图及其对应的 $\mathbf{M}_{\text{view}}^i, i \in \{1, 2\}$ 。本文首先采用经典的基于 ORB 的特征点匹配算法^[29], 对两幅视图上的点进行匹配。由于匹配的点对来自于世界坐标下的同一点,因此有

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = w_c^i (\mathbf{M}_{\text{view}}^i)^{-1} (\mathbf{M}_{\text{project}})^{-1} (\mathbf{M}_{\text{viewport}})^{-1} \begin{bmatrix} u^i \\ v^i \\ z_{\text{NDC}}^i \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2u^i - W}{Wf_1} R_{11}^i + \frac{2v^i - H}{Hf_2} R_{12}^i + R_{13}^i \\ \frac{2u^i - W}{Wf_1} R_{21}^i + \frac{2v^i - H}{Hf_2} R_{22}^i + R_{23}^i \\ \frac{2u^i - W}{Wf_1} R_{31}^i + \frac{2v^i - H}{Hf_2} R_{32}^i + R_{33}^i \\ \frac{z_{\text{NDC}}^i}{c_2} + \frac{c_1}{c_2} \end{bmatrix}, i \in \{1, 2\} \quad (8)$$

$$\begin{bmatrix} u^2 \\ v^2 \\ z_{\text{NDC}}^2 \\ 1 \end{bmatrix} = \frac{w_c^1}{w_c^2} \mathbf{M} \begin{bmatrix} u^1 \\ v^1 \\ z_{\text{NDC}}^1 \\ 1 \end{bmatrix}$$

$$\mathbf{M} = \mathbf{M}_{\text{viewport}} \mathbf{M}_{\text{project}} \mathbf{M}_{\text{view}}^2 (\mathbf{M}_{\text{view}}^1)^{-1} (\mathbf{M}_{\text{project}})^{-1} (\mathbf{M}_{\text{viewport}})^{-1} \quad (5)$$

式中:上标 1 和 2 分别表示两幅视图对应的参数信息。实际上,由于大部分商业化电子游戏平台都不开源,相比于游戏开发者可以很容易地从游戏引擎中拿到图像渲染上下文,用户往往很难获得相机的内参进行上述视图间的变换。因此,如何利用已知视图对相机内参进行一定程度的估计,对有效利用游戏中的三维投影变换是至关重要的。

2 基于三维投影变换的内参估计算法

本文在相机旋转下,基于三维投影变换对相机内参进行了一定的估计。依据式(2),假设待估计的相机投影矩阵 $\mathbf{M}_{\text{project}}$ 为

$$\mathbf{M}_{\text{project}} = \begin{bmatrix} f_1 & 0 & 0 & 0 \\ 0 & f_2 & 0 & 0 \\ 0 & 0 & c_1 & c_2 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (6)$$

式中: f_1, f_2, c_1, c_2 分别对应于投影矩阵中的 4 个参数。相机旋转下的 $\mathbf{M}_{\text{view}}$ 可表示为

$$(\mathbf{M}_{\text{view}}^i)^{-1} (\mathbf{M}_{\text{project}})^{-1} (\mathbf{M}_{\text{viewport}})^{-1} =$$

$$\begin{bmatrix} R_{11}^i & R_{12}^i & R_{13}^i & 0 \\ R_{21}^i & R_{22}^i & R_{23}^i & 0 \\ R_{31}^i & R_{32}^i & R_{33}^i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{f_1} & 0 & 0 & 0 \\ 0 & \frac{1}{f_2} & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{1}{c_2} & \frac{c_1}{c_2} \end{bmatrix}.$$

$$\begin{bmatrix} \frac{2}{W} & 0 & 0 & -1 \\ 0 & \frac{2}{H} & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

结合式(5),可对 f_1, f_2 建立 3 个约束

$$\begin{cases} A = \frac{((2u^1 - W)/Wf_1)R_{11}^1 + ((2v^1 - H)/Hf_2)R_{12}^1 + R_{13}^1}{((2u^2 - W)/Wf_1)R_{11}^2 + ((2v^2 - H)/Hf_2)R_{12}^2 + R_{13}^2} = \frac{w_c^2}{w_c^1} \\ B = \frac{((2u^1 - W)/Wf_1)R_{21}^1 + ((2v^1 - H)/Hf_2)R_{22}^1 + R_{23}^1}{((2u^2 - W)/Wf_1)R_{21}^2 + ((2v^2 - H)/Hf_2)R_{22}^2 + R_{23}^2} = \frac{w_c^2}{w_c^1} \\ C = \frac{((2u^1 - W)/Wf_1)R_{31}^1 + ((2v^1 - H)/Hf_2)R_{32}^1 + R_{33}^1}{((2u^2 - W)/Wf_1)R_{31}^2 + ((2v^2 - H)/Hf_2)R_{32}^2 + R_{33}^2} = \frac{w_c^2}{w_c^1} \end{cases} \quad (10)$$

通过 $\min((A-B)^2 + (C-B)^2 + (A-C)^2)$, 采用经典的非线性最小二乘法便可求解得到 f_1 以及 f_2 。考虑到游戏视频, 尤其是第一人称游戏视频, 往往包含大量的相机旋转运动, 采用本文算法可以很容易估计出 f_1 以及 f_2 。为提升匹配的准确性, 本文采用纹理相对丰富的已知视图进行匹配。此外需要注意的是, 当相机仅绕 x 轴或 y 轴旋转时, 目标函数仅为 f_2 或 f_1 的相关函数, 因此采用多轴复合旋转运动可以更加方便快捷地计算出 f_1 、 f_2 。

基于此, 通过抓取游戏视频中相机旋转的两帧, 采用本文提出的基于三维投影变换的相机内参估计算法, 便可通过上述过程求得相机投影矩阵的 f_1 、 f_2 参数。利用图像单应性^[30], 可在已知相机运动信息的前提下, 仅利用 f_1 、 f_2 实现对已知视图的旋转变换。不同坐标表示的其他游戏引擎内参也可通过类似方法得到。

3 三维投影变换辅助云游戏编码优化

在游戏序列中, 视角通常会保持移动, 尤其是在第一人称游戏中, 相机的运动往往是旋转角度更大、移动频率更高的旋转和平移的复合运动, 然而传统运动估计对相机旋转较为敏感。本节利用第2节中基于三维投影变换的相机内参估计算法, 结合图像单应性特点, 利用视图间的旋转信息生成与当前待编码帧平行的辅助参考帧, 然后将其添加到编码器的参考帧列表中参与编码。由于辅助参考帧利用相机的旋转信息进行了视角校正, 与待编码帧之间只有平移, 因此更有利于进行运动估计, 从而减少编码残差, 优化编码性能。

3.1 辅助参考帧的生成

相机在发生旋转前后呈现的两幅视图具有图像单应性^[30], 因此相机旋转运动下从图像1到图像2的映射可由式(5)简化为

$$\begin{cases} \mathbf{P}_{\text{pixel}}^2 = \mathbf{H}\mathbf{P}_{\text{pixel}}^1 \\ \mathbf{H} = \mathbf{V}\mathbf{K}\mathbf{R}^2(\mathbf{R}^1)^{-1}\mathbf{K}^{-1}\mathbf{V}^{-1} \end{cases} \quad (11)$$

式中: $\mathbf{H}$ 表示 3×3 的单应性矩阵; $\mathbf{K}$ 、 $\mathbf{V}$ 分别表示 3×3 的相机内参矩阵(投影矩阵)和视口变换矩阵。

由第1节中 $\mathbf{M}_{\text{viewport}}$ 和 $\mathbf{M}_{\text{project}}$ 的表示形式, 不难得到相机旋转下的 $\mathbf{V}$

$$\mathbf{V} = \begin{bmatrix} \frac{W}{2} & 0 & \frac{W}{2} \\ 0 & \frac{H}{2} & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

相机旋转下的 $\mathbf{K}$ 可表示为

$$\mathbf{K} = \begin{bmatrix} \frac{\cot(V'/2)}{A} & 0 & 0 \\ 0 & \cot(V'/2) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (13)$$

且其参数可通过第2节的内参估计算法得到。

因此, 给定参考视图与相机旋转矩阵, 即可依据式(11)在不使用深度图的情况下实现从参考视图到任意目标视图的空间映射。基于此, 本文选择每个编码视频序列(CVS)中质量较高的帧内随机接入点(IRAP)帧作为映射过程的参考帧, 利用上述公式通过图像单应性校正相机旋转, 并应用三角形重心插值算法^[31]模拟游戏引擎的渲染过程, 生成与当前CVS中其他待编码帧平行的辅助参考帧。由于旋转之后视图之间只有平移, 本文算法更有利于进行运动估计, 从而减少压缩残差所需的比特。此外, 当游戏序列包含单一的相机旋转运动时, 生成的辅助参考帧与待编码帧同视点同方向, 可以最大程度地减小编码残差。相比于传统编码器, 本文算法额外增加的比特开销只有相机的旋转信息与两个内参, 且内参通常对整个序列保持相同并且只需要传输一次。

3.2 参考帧列表的更新

利用生成的辅助参考帧更新编码器的参考帧列表, 以提供更好的运动估计和补偿。以标准 Low-Delay-P (LDP) 配置下 HEVC 参考软件 HM 为例, 参考帧列表最多包含 4 个以图像序列号(POC, 用符号 I_{POC} 表示)从大到小顺序排列的参考帧。本文将生成的辅助参考帧添加到原始参考帧列表的第二个位置, 第一个位置的原始参考帧保持不变以保持较强的相关性, 除最后一帧外的其余原始参考帧按顺序依次排列在辅助参照帧之后。更新过程见图2。辅助参考帧的 POC 设为 -1, 与原始参考帧进行区分。如果原始参考帧列表包含 POC 为 3、2、1、0 的参考帧, 则更新后参考帧的 POC 依次为 3、-1、2、1。

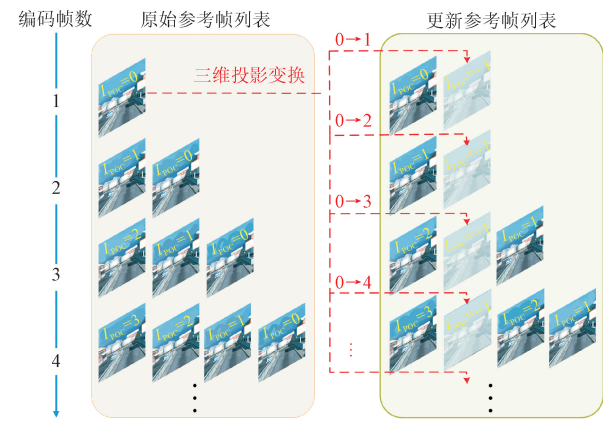


图 2 参考帧列表更新示意

Fig. 2 Schematic of updating the reference frame list

4 实验与分析

为验证所提云游戏编码优化算法的有效性,将本文算法加入到 HEVC 参考软件 HM16.16 中,从编码性能和编码复杂度两方面与原有的 HEVC 编码算法进行对比。

4.1 实验设置

由于云游戏低时延的要求,本文以标准 LDP 为配置,在 Intel (R) Core (TM) i7-9700KF CPU @ 3.60 GHz 处理器上,以 4 个量化参数 22、27、32、37 进行编码。参数 IntraPeriod、intrqpoffset 以及 GOPSize 分别设置为 8、0 和 4,编码帧数设置为 96。本文选择编码序列中每个 CVS 中的 IRAP 帧作为映射参考帧,用以生成优化算法中相同 CVS 中其他帧的辅助参考帧。

由于标准测试序列大部分为不含相机运动信息的自然序列,本文使用 Air Light VR(ALVR)软件将 SteamVR 游戏流式传输到客户端独立的 VR 头戴式显示器中,通过捕获视频帧及其对应的相机运动信息,建立包含渲染图像和相机运动的云游戏视频测试序列。本文将 VR 游戏的里程碑之作 The Lab 作为游戏数据库,选取其中具有不同内容特征的 5 个场景下的 20 个 YUV 8 bit 序列作为测试序列。每个场景下的序列分别包含 4 种不同的大幅相机运动方式,每个序列的分辨率为 1 024×1 024。图 3 为 5 个场景的示意。

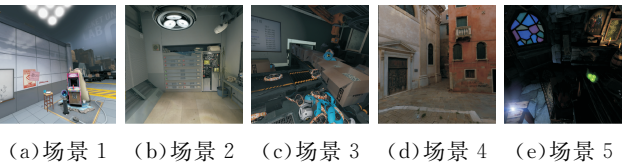


图 3 游戏场景示意

Fig. 3 Schematic of gaming scenes

4.2 编码性能

本小节首先通过 Bjøntegaard-Delta rate(BD-rate, 用符号 ΔR 表示)和 Bjøntegaard-Delta PSNR (BD-PSNR,用符号 ΔP 表示)^[32] 评估所提优化算法在 YUV 上的编码性能。BD-rate 表示在相同编码质量下节省了多少码率,负值表示所提算法具有更高的编码效率;BD-PSNR 表示在相同码率下可以获得的质量增益,正值表示所提算法具有更优性能。

将本文算法与原始 HEVC 进行比较,结果如表 1 所示。相比 HEVC,本文算法额外的比特增加只有相机的旋转信息与两个内参,因此本文在表 1 的计算中忽略这部分开销。

表 1 本文算法与 HEVC 在 5 个场景 4 种相机运动下压缩性能的比较

Table 1 Comparison of the proposed algorithm with HEVC in compression performance of four camera movements in five scenes

相机运动	场景编号	$\Delta R_{\text{HEVC}}/\%$	$\Delta P_{\text{HEVC}}/\text{dB}$
前进旋转	1	-2.04	0.093 0
	2	-1.77	0.054 3
	3	-1.52	0.060 5
	4	-1.35	0.036 8
	5	-5.77	0.212 2
	均值	-2.49	0.091 3
后退旋转	1	-1.93	0.085 7
	2	-1.47	0.054 9
	3	-1.32	0.051 6
	4	-3.72	0.109 4
	5	-3.04	0.108 6
	均值	-2.30	0.082 0
平移旋转	1	-2.04	0.090 9
	2	-6.05	0.140 6
	3	-2.06	0.078 7
	4	-3.58	0.089 3
	5	-1.82	0.063 6
	均值	-3.11	0.092 6
单独旋转	1	-22.05	1.026 3
	2	-23.51	0.818 1
	3	-22.31	0.933 6
	4	-29.96	1.129 5
	5	-25.60	1.038 5
	均值	-24.68	0.989 2
总均值		-8.14	0.313 8

从表 1 可知,从均值来看,相比于原始 HEVC,

本文算法在相同视频客观质量下可以节省 8.14% 的比特率,在相同的码率下可以实现 0.313 8 dB 的性能增益。针对不同的相机运动方式,本文算法都有良好的编码性能增益。其中,在相机单独旋转运动下,BD-rate 和 BD-PSNR 性能可达-24.68%和 0.989 2 dB。这是因为在单一的相机旋转运动下,本文算法生成的辅助参考帧与待编码帧同视点同方向,最大程度地减小了编码残差,提升了编码性能。此外,相机运动的幅度与快慢也会影响算法的性能。例如,在单独旋转下,相比于其他 4 个场景,场景 4 的相机旋转幅度更大,变化更剧烈,原始 HEVC 无法对如此大范围的运动进行很好的运动估计和补偿,而本文算法克服了运动范围与幅度的限制,利用三维投影变换较为准确地捕获了大运动下视图间像素的运动,因此可以获得-29.96%的 BD-rate 性能。综上可知,本文算法在不同游戏场景不同运动下都可以实现良好的压缩性能。

为了对编码后的视频进行进一步的视觉客观评价,表 2 给出了量化参数(QP)为 22、27、32、37 编码后的视频结构相似度(structural similarity, SSIM)和多尺度结构相似度值(multi-scale structural similarity, MS-SSIM)。可以看出,本文算法编码后的视频相比于 HEVC 仍然没有视频质量的降低。在达到相同客观质量的情况下,本文算法所需的码率更低,本文算法的优越性得到了进一步验证。

表 2 4 个 QP 编码后视频的结构相似度和多尺度结构相似度

Table 2 SSIM and MS-SSIM of encoded videos under four QPs

量化参数	结构相似度		多尺度结构相似度	
	本文算法	HEVC	本文算法	HEVC
22	0.973 4	0.973 3	0.994 0	0.994 0
27	0.955 2	0.954 5	0.987 0	0.986 7
32	0.931 8	0.930 3	0.974 0	0.973 2
37	0.903 1	0.900 8	0.951 8	0.949 7
均值	0.940 9	0.939 7	0.976 7	0.975 9

同时,考虑到云游戏是提供计算机生成文本、图形和动画等屏幕内容的典型应用之一,本文算法与 HEVC 屏幕内容编码扩展(HEVC screen content coding, HEVC-SCC)^[15]进行了对比,结果如表 3 所示。可以看出,尽管 HEVC-SCC 可以利用云游戏视频具有重复图案、高度饱和度或者有限数量颜色等不同于自然视频的特性对其实现良好的压缩,却无法应对频繁视角变化的情况。相比之下,本文算

法充分考虑了云游戏的应用特性,实现了任意视角与运动下良好的压缩性能,相比于 HEVC-SCC 有超过 20% 的 BD-rate 性能增益,从而进一步证明了本文算法相比于目前云游戏应用中主流视频编码方式的优越性。

表 3 本文算法与 HEVC-SCC 平均压缩性能的比较
Table 3 Comparison of the proposed algorithm with HEVC-SCC in average compression performance

场景编号	$\Delta R_{\text{HEVC-SCC}}/\%$	$\Delta P_{\text{HEVC-SCC}}/\text{dB}$
1	-21.02	1.038 8
2	-18.63	0.747 0
3	-21.56	0.978 9
4	-27.09	0.964 8
5	-26.14	1.112 6
均值	-22.89	0.968 4

由于本文算法与文献[28]算法都利用了相机运动信息来优化编码器的参考帧列表,以此提高云游戏视频的编码效率,所以本文算法与文献[28]算法进行了比较,亮度分量上的压缩性能结果如表 4 所示。由于文献[28]算法仅针对相机旋转下的视频序列进行了训练,所以仅对单独旋转下的序列采用文献[28]中训练好的模型进行测试。可以看出,本文算法在 5 个场景下的压缩性能都远远高于文献[28]算法,平均可以实现-24.12%的 BD-rate 和 1.005 0 dB 的 BD-PSNR 性能增益。这是因为本文的测试序列运动幅度更大,视角变换更加剧烈,采用文献[28]中增强参考帧的方法已经无法从根本上对如此剧烈变化的运动实现更好的运动估计与补偿效果,而本文算法利用三维投影变换生成与当前待编码帧更相似的辅助参考帧,进一步降低了剧烈运动下运动估计的难度,且大幅度减少了编码残差,从而实现了良好的压缩效果。

表 4 本文算法与文献[28]算法压缩性能的比较
Table 4 Comparison of the proposed algorithm with Ref. [28] in compression performance

场景编号	$\Delta R_{\text{ref}}/\%$	$\Delta P_{\text{ref}}/\text{dB}$
1	-21.99	1.060 6
2	-23.03	0.883 8
3	-21.97	0.948 4
4	-29.08	1.123 3
5	-24.54	1.009 0
均值	-24.12	1.005 0

4.3 辅助参考帧有效性

生成的辅助参考帧、原始参考帧以及待编码帧如图 4 所示。这里原始参考帧指相关性最强的前一帧的重构帧。可以明显看出,生成的辅助参考帧实现了视角的校正,相比于原始参考帧,与待编码帧之间只有简单的平移运动,更容易进行运动估计。利用该辅助参考帧可以使编码器更好地估计云游戏视频中复杂的全局相机旋转以及剧烈变化的大范围运动,提高云游戏的编码效率。



图 4 辅助参考帧与原始参考帧及待编码帧的对比
Fig. 4 Comparison of the auxiliary reference frame with the original reference frame and the frame to be encoded

各场景下,预测块采用辅助参考帧进行预测的辅助参考帧命中率如图 5 所示。可以看出:生成的辅助参考帧深度参与了编码,其命中率随运动状态的不同而变化,最高可超过 50%;辅助参考帧的命中率呈规律性分布,每个 CVS 中距离映射参考帧(即 IRAP 帧)越近的帧,采用辅助参考帧进行预测的命中率越高。这是由于距离较远的帧与映射参考帧之间的场景重复度降低,大量未在映射参考帧中出现的新场景逐渐暴露出来,导致生成的辅助参考帧与待编码帧相似度降低,从而降低了命中率。

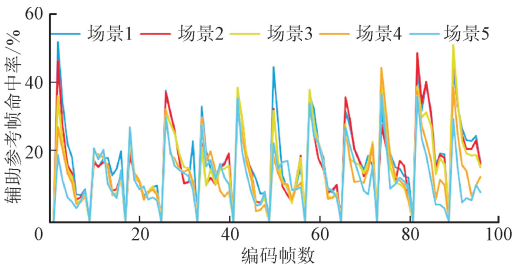


图 5 使用辅助参考帧进行预测的比例
Fig. 5 Proportion of predictions using auxiliary reference frames

4.4 编码复杂度

设 T_{HEVC} 为原始 HEVC 的编码时间, $T_{proposed}$ 为本文算法的编码时间(包括额外生成辅助参考帧的时间以及将辅助参考帧添加到参考帧列表进行编码的时间),则编码复杂度衡量指标 ΔT 可表示为

$$\Delta T = (T_{proposed} - T_{original}) / T_{original} \quad (14)$$

ΔT 为负表明本文算法编码复杂度较低。表 5 给出了 4 种运动方式下编码时间的对比。可以看出,相比原始 HEVC,本文算法在带来大幅度性能增益的同时还在一定程度上降低了编码复杂度,编码时间平均缩短 5.34%。

表 5 本文算法与 HEVC 编码时间的比较
Table 5 Comparison of the proposed algorithm with HEVC in encoding time

相机运动	$T_{proposed} / s$	T_{HEVC} / s	$\Delta T / \%$
前进旋转	952.366	1 020.041	-6.63
后退旋转	981.742	1057.708	-7.18
平移旋转	931.410	958.177	-2.79
单独旋转	887.971	929.339	-4.45
均值	938.372	991.316	-5.34

5 结 论

随着高性能云计算等技术的快速发展,云游戏生态逐步扩大,高帧率以及高分辨率的云游戏视频对传输带宽提出了更高的要求,然而云游戏视频中复杂的全局相机旋转以及剧烈变化的大范围运动给传统视频压缩带来了极大的挑战。本文针对这一问题,利用云游戏视频易于获得全局相机运动信息这一特点,提出了一种三维投影变换辅助的云游戏编码优化算法。

本文通过对游戏场景中三维投影变换的过程进行理论分析,利用来自世界坐标下同一点的匹配点对建立约束,构建了虚拟相机的内参估计算法。结合全局相机运动信息,利用估计的内参生成与待编码帧平行的辅助参考帧更新原始参考帧列表,参与编码。由于生成的辅助参考帧相对于待编码帧只有平移,更有利于进行运动估计和补偿。本文算法克服了开源云游戏平台限制,使利用三维投影变换对云游戏编码优化变得一般化。实验结果表明,本文算法在云游戏视频上平均可以实现 8.14% 的 BD-rate 增益,在仅有相机旋转的情况下,BD-rate 可达-24.68%。同时,还在一定程度上降低了编码复杂度,编码时间平均缩短 5.34%。此外,本文算法相比 HEVC-SCC 也有超过 20% 的 BD-rate 性能增益。

云游戏的大规模应用对通信系统提出了多项挑战,除高带宽外,云游戏对网络时延要求很高,尤其在实时竞技类游戏中,卡顿现象会严重影响用户体

验。因此,除了提高云游戏视频的编码效率以降低带宽外,如何满足云游戏对网络的低时延要求也在进一步研究中。

参考文献:

- [1] CAI Wei, SHEA R, HUANG Chunying, et al. A survey on cloud gaming: future of computer games [J]. IEEE Access, 2016, 4: 7605-7620.
- [2] DI DOMENICO A, PERNA G, TREVISAN M, et al. A network analysis on cloud gaming: stadia, GeForce now and PSNow [J]. Network, 2021, 1(3): 247-260.
- [3] WU Jiyang, YUEN C, CHEUNG N M, et al. Streaming mobile cloud gaming video over TCP with adaptive source: FEC coding [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 27(1): 32-48.
- [4] JARSCHER M, SCHLOSSER D, SCHEURING S, et al. An evaluation of QoE in cloud gaming based on subjective tests [C]//2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. Piscataway, NJ, USA: IEEE, 2011: 330-335.
- [5] LE H, POURREZA R, SAID A, et al. GameCodec: neural cloud gaming video codec [C]//The 33rd British Machine Vision Conference. Durham, UK: BMVA Press, 2022: 204.
- [6] SULLIVAN G J, OHM J R, HAN W J, et al. Overview of the high efficiency video coding (HEVC) standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1649-1668.
- [7] NORKIN A, BJONTEGAARD G, FULDSETH A, et al. HEVC deblocking filter [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1746-1754.
- [8] HUANG Yuwen, HSU C W, CHEN C Y, et al. A VVC proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(5): 1311-1325.
- [9] GAO Han, ESENLIK S, ALSHINA E, et al. Geometric partitioning mode in versatile video coding: algorithm review and analysis [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 31(9): 3603-3617.
- [10] 李维, 张和仙, 杨付正. 高效率视频编码快速模式判决算法 [J]. 西安交通大学学报, 2013, 47(8): 104-109, 140.
- LI Wei, ZHANG Hexian, YANG Fuzheng. A fast mode decision algorithm for high efficiency video coding [J]. Journal of Xi'an Jiaotong University, 2013, 47(8): 104-109, 140.
- [11] TISSIER A, HAMIDOUCE W, VANNE J, et al. CNN oriented complexity reduction of VVC intra encoder [C]//2020 IEEE International Conference on Image Processing (ICIP). Piscataway, NJ, USA: IEEE, 2020: 3139-3143.
- [12] 公衍超, 吕显龙, 杨楷芳, 等. 考虑全帧间参考依赖性的监控视频高效视频编码码率控制算法 [J]. 西安交通大学学报, 2021, 55(3): 72-80.
- GONG Yanchao, LÜ Xianlong, YANG Kaifang, et al. Rate control for high efficiency video coding coded surveillance videos considering all of the inter-frame reference dependency [J]. Journal of Xi'an Jiaotong University, 2021, 55(3): 72-80.
- [13] 李维, 任鹏, 赵凡, 等. 高效视频编码中帧内码率控制改进算法 [J]. 西安交通大学学报, 2019, 53(4): 79-84.
- LI Wei, REN Peng, ZHAO Fan, et al. A modified method for intra-coding rate control in high efficiency video coding [J]. Journal of Xi'an Jiaotong University, 2019, 53(4): 79-84.
- [14] XU Jizheng, JOSHI R, COHEN R A. Overview of the emerging HEVC screen content coding extension [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 26(1): 50-62.
- [15] XU Xiaozhong, LIU Shan. Overview of screen content coding in recently developed video coding standards [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(2): 839-852.
- [16] XU Lingfeng, GUO Xun, LU Yan, et al. A low latency cloud gaming system using edge preserved image homography [C]//2014 IEEE International Conference on Multimedia and Expo (ICME). Piscataway, NJ, USA: IEEE, 2014: 1-6.
- [17] METZGER F, GEIBLER S, GRIGORJEW A, et al. An introduction to online video game QoS and QoE influencing factors [J]. IEEE Communications Surveys & Tutorials, 2022, 24(3): 1894-1925.
- [18] ZADTOOTAGHAJ S, SCHMIDT S, MÖLLER S. Modeling gaming QoE: towards the impact of frame rate and bit rate on cloud gaming [C]//2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX). Piscataway, NJ, USA: IEEE, 2018: 1-6.
- [19] ZADTOOTAGHAJ S, SCHMIDT S, SABET S S, et al. Quality estimation models for gaming video streaming services using perceptual video quality dimensions [C]//Proceedings of the 11th ACM Multimedia Sys-

- tems Conference. New York, NY, USA: Association for Computing Machinery, 2020: 213-224.
- [20] AHMADI H, ZAD TOOTAGHAJ S, HASHEMI M R, et al. A game attention model for efficient bit rate allocation in cloud gaming [J]. *Multimedia Systems*, 2014, 20(5): 485-501.
- [21] REN Guangjie, LIU Feiyang, WANG Huairui, et al. Multi-objective optimization based perceptual bit allocation for gaming video coding in VVC [J]. *Signal Processing*, 2022, 198: 108593.
- [22] SEMSARZADEH M, YASSINE A, SHIRMOHAMMADI S. Video encoding acceleration in cloud gaming [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2015, 25(12): 1975-1987.
- [23] HEGAZY M, DIAB K, SAEEDI M, et al. Content-aware video encoding for cloud gaming [C]//Proceedings of the 10th ACM multimedia systems conference. New York, NY, USA: Association for Computing Machinery, 2019: 60-73.
- [24] SHI Shu, HSU C H, NAHRSTEDT K, et al. Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming [C]//Proceedings of the 19th ACM international conference on Multimedia. New York, NY, USA: Association for Computing Machinery, 2011: 103-112.
- [25] LEE K, CHU D, CUERVO E, et al. Outatime: using speculation to enable low-latency continuous interaction for mobile cloud gaming [C]//Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services. New York, NY, USA: Association for Computing Machinery, 2015: 151-165.
- [26] LI Yong, GAO Wei. DeltaVR: achieving high-performance mobile VR dynamics through pixel reuse [C]//Proceedings of the 18th International Conference on Information Processing in Sensor Networks. New York, NY, USA: Association for Computing Machinery, 2019: 13-24.
- [27] KIM H D, LEE J W, OH T W, et al. Robust DT-CWT watermarking for DIBR 3D images [J]. *IEEE Transactions on Broadcasting*, 2012, 58(4): 533-543.
- [28] WANG Yifan, WANG Hao, WANG Kaijie, et al. Cloud gaming video coding optimization based on camera motion-guided reference frame enhancement [J]. *Applied Sciences*, 2022, 12(17): 8504.
- [29] RUBLEE E, RABAUD V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF [C]//2011 International Conference on Computer Vision. Piscataway, NJ, USA: IEEE, 2011: 2564-2571.
- [30] LIN Bin, XU Xiangpeng, SHEN Zhihua, et al. A registration algorithm for astronomical images based on geometric constraints and homography [J]. *Remote Sensing*, 2023, 15(7): 1921.
- [31] DZIEMBOWSKI A, GRZELKA A, MIELOCH D, et al. Multiview synthesis: improved view synthesis for virtual navigation [C]//2016 Picture Coding Symposium (PCS). Piscataway, NJ, USA: IEEE, 2016: 1-5.
- [32] BJONTEGAARD G. Calculation of average PSNR differences between RD-curves [EB/OL]. (2001-03-26) [2023-04-27]. https://www.itu.int/wftp3/avarch/video-site/0104_Aus/VCEG-M33.doc.

(编辑 陶晴)