

《FoV-Aware Edge Caching for Adaptive 360° Video Streaming》 - Review

CUC - 19 数媒技 杨雪婷

2021.04.01

汇报内容介绍

1. 问题

2. 结构

3. 算法

4. 实验

Part One : 问题

缓存基于贴图的360视频的挑战是

- (i)如何用适当分辨率的贴图填充有限大小的缓存，以及适当的分辨率。
- (ii)当缓存容量超过时，将**哪些贴图从缓存中驱逐出来**，以便为新的贴图留出空间。

Part Two : 结构

客户端根据FoV和网络条件来决定tile的请求质量（高 / 低）。

Fov Prediction:

每100ms从客户端提取100样本

选出前10个最新请求的为输入，利用WLR 预测1s后的Fov。

Network:

两种策略:

- 1、 Fov 内用高分辨率, 其他用低分辨率, $biterate : [14\% \cdot R_{\text{high}} + 86\% \cdot R_{\text{low}}]$
- 2、 对所有的分块都请求低分辨率, $biterate : [100\% \cdot R_{\text{low}}]$

throughput 样本的计算方法:

$$throughput_sample = \frac{\sum_{j=1}^t size(T_j)}{\sum_{j=1}^t download_time(T_j)}$$

network throughput estimation = 最近3个样本的平均值

如果 $network$ 在 $\left[\frac{f}{t} \cdot R_{high} + \frac{t-f}{t} \cdot R_{low} \right]$ 区间内就可以第一种策略选择分辨率

Part Three : 算法

360°视频的前提特性

(1) FoV Pattern与内容的关系 -- heatmap 表示 -- 得到 common-Fov:

- 博物馆观光类：头部转动较多但是轨迹相似
- 赛车比赛类：较激烈的运动在前方，很少有向后看的

(2) Impact of Video's Bitrate on FoV's Quality

网络条件固定时，Video's Bitrate \uparrow , FoV's Quality \downarrow 。

提出 FoV-aware Caching Policy

adaption logic -- 根据计算的network 情况选择以何种方式缓存

Policy 考虑两个指标 (based on viewing history) :

- 1、 $tile$ 在 $common - Fov$ 中的概率 $\theta_{v,s,i}$
- 2、 用户要求每个视频在 Fov 中的 $tile$ 以高质量请求的概率 ψ_v

通过**最大似然估计**后求导得到上面两个参数。

$$\theta_{v,s,i} = \frac{\alpha_{inFoV_{v,s,i}}}{\alpha_{inFoV_{v,s,i}} + \alpha_{outOfFoV_{v,s,i}}}$$

$$\psi_v = \frac{\beta_{high_v}}{\beta_{high_v} + \beta_{low_v}}$$

进而通过 $\gamma_{v,s,i,q} = \begin{cases} \theta_{v,s,i} * \psi_v, & \text{if } q = \text{high} \\ (1 - \theta_{v,s,i}) + \theta_{v,s,i} * (1 - \psi_v), & \text{if } q = \text{low} \end{cases}$ 得到 $\gamma_{v,s,i,q}$

γ 代表：tile t 将在未来以q的质量请求的概率。

γ 越小越优先剔除

Part Four : 实验

基本状况:

24 tiles, 500条轨迹, 1s30帧, 1s一个视频segment

bitrate:

高质量: 640×540

低质量: 320×270

cache capacity: 25% | 50% of the total

比较策略: LRU | LFU

评价指标：

- *Tile hit* (缓存命中率)
- *Bandwidth saving*
- *Reb*(缓冲频率)：没有任何视频可以播放的频率
- *DoR* [延迟时间(s)]:该指标显示客户端经历延迟的总持续时间。
- *Qua*[高质量的FoV(以百分比表示)]