High-Quality Immersive Video Streaming
via Edge Caching and User Adaptation

Jinlei Jiang
Tsinghua University, China
jjle@tsinghua.edu.cn

Joint work with: Dongbiao He, Cedric Westphal, Guangwen Yang, JJ Garcia-Luna-Aceves, Teng Ma, Shu-Tao Xia
About Me

• Associate Professor of Computer Science and Technology, Tsinghua University, P. R. China
• Humboldt Research Fellow (2007-2008)

Research Interests
• distributed computing and systems
• big data storage and computing
• cloud/edge computing
• graph computing and database
• software-defined networking

http://madsys.cs.tsinghua.edu.cn/~jinleijiang/
Outline

• Background & Challenges
• User Behavior Analysis
• Edge Caching & Prefetching
• Evaluation
• Concluding Remarks
Immersive video is popular now!

Immersive video, a.k.a 360-degree or spherical video, can provide users with immersive and interactive experience under their own control.

Record: 360 camera  View: HMD or Glasses  Education  Games  Business

Wide applications in various domains

$47.7B  https://www.mordorintelligence.com/industry-reports/virtual-reality-market/

The global market of immersive video streaming would reach by 2024
An overview of the video streaming system

Images from camera → Video frames at server → Stitched images shown for users
Challenges of streaming immersive videos

large storage need
- Store **multiple views** of each scene for a large variety of client devices
- Keep video **resolution high** for good experience

high BW consumption
- At least **4K stream** is needed to transmit a video in full view
- Serve **many users** at the same time

ultralow motion-to-photon delay
- The new view must be rendered in **very limited time** for good experience

- 3GB/minutes in size
- 400Mbps
  - 25Mbps (2D 4K video)
- < 10 milliseconds

Refer to MICHAEL ZINK et al., PROCEEDINGS OF THE IEEE, Vol. 107, No. 4, for more!
Practice: User/FoV adaptation

FoV: Field of View $\leftrightarrow$ Viewport
Practice: User/FoV adaptation (cont.)

Viewport-Driven Rate-Distortion Optimized 360° Video Streaming
Jacob Chakareski, Ridvan Aksu, Xavier Corbillion, Gwendal Simon, and Viswanathan Swaminathan

FoV-Aware Edge Caching for Adaptive 360° Video Streaming
Anahita Mahzari, Afshin Taghavi Nasrabadi, Aliehsan Samiei and Ravi Prakash
The University of Texas at Dallas

View-Aware Tile-Based Adaptations in 360 Virtual Reality Video Streaming
Mohammad Hosseini*

Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices
Feng Qian1*, Bo Han2, Qingyang Xiao1, Vijay Gopala1
1Indiana University 2AT&T Labs – Research

Joint Rate and FoV adaptation in immersive video streaming
Dongbiao He
University of California, Santa Cruz
Santa Cruz, CA
dongbiosa@ucsc.edu

Two-Layer FoV Prediction Model for Viewport Dependent Streaming of 360-Degree Videos
Yiling Xu(✉), Shaowei Xie, Liangji Ma, and Jun Sun

CUB360: EXPLOITING CROSS-USER BEHAVIORS FOR VIEWPORT PREDICTION IN 360 VIDEO ADAPTIVE STREAMING
Yuxuan Ban1, Lan Xie1, Zhimin Xu1, Xinggong Zhang1,2,* Zongming Guo1,2, Yue Wang3
1Indiana University 2AT&T Labs – Research
Practice: In-network caching

• An old yet new idea

• Questions to answer
  • Where to place the cache & what’s the unit (video or tile) for caching
  • How to adapt the bitrate according to network condition

• A lot of work
  • FoV-aware edge caching (MM’18), tile-based caching (MobiHoc’19), JERTC (MMM’19), Allies (Cloud’20), ...
Our solution: CUBIST

\[
\text{CUBIST} = \text{Edge Caching (video popularity)} + \text{Tile Prefetching (FoV predication)}
\]
Outline

• Background & Challenges
• User Behavior Analysis
• Edge Caching & Prefetching
• Evaluation
• Concluding Remarks
Datasets, analysis method and focus

**Dataset:** Xavier Corbillon et al., MMSys’17

- Exploration (Paris)
- Static Focus (Rhino)
- Rides (Rollercoaster)
- Moving Focus (Timelapse)

**characteristics:**
- It collects the user head movement data
- The dataset contains **59 users**
- Multiple kinds of videos: **6 videos**

**Method:** projection and tiling

**Focus:**
- Viewer motion
- Head movement

- Raw Data Analysis
- Tiling
- Viewport Variation
- Tile Transition
- Tile Interval
Result: raw data analysis

User’s eye position is **hard to predict** especially in long time

Conclusion: it is **not useful** to directly estimate the eye position
Result: viewport is predictable

(1) User moves **shortly** during a given interval:
- e.g., 85% of users moves 0.956 unit within 1000ms

(2) Only **part of the view** (FoV) needed by the client
- e.g., uses less than 30.4% of the view in the sphere

<table>
<thead>
<tr>
<th></th>
<th>100ms</th>
<th>250ms</th>
<th>500ms</th>
<th>750ms</th>
<th>1000ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>95%</td>
<td>0.147</td>
<td>0.433</td>
<td>3.012</td>
<td>3.093</td>
<td>3.107</td>
</tr>
<tr>
<td>90%</td>
<td>0.096</td>
<td>0.255</td>
<td>0.567</td>
<td>1.11</td>
<td>2.983</td>
</tr>
<tr>
<td>85%</td>
<td>0.073</td>
<td>0.19</td>
<td>0.401</td>
<td>0.645</td>
<td>0.956</td>
</tr>
</tbody>
</table>
Result: tile request distribution

Key Findings:
1. Only a small portion of tiles are requested by users;
2. The tile frequency varies greatly inside a video;
3. Most kinds of videos show the same behavior, while some other videos are not.

<table>
<thead>
<tr>
<th>Tiling</th>
<th>Paris</th>
<th>Rhino</th>
<th>Rollercoaster</th>
<th>Timelapse</th>
</tr>
</thead>
<tbody>
<tr>
<td>6*8</td>
<td>0.35</td>
<td>0.35</td>
<td>0.31</td>
<td>0.40</td>
</tr>
<tr>
<td>9*12</td>
<td>0.34</td>
<td>0.32</td>
<td>0.31</td>
<td>0.36</td>
</tr>
<tr>
<td>12*12</td>
<td>0.32</td>
<td>0.28</td>
<td>0.31</td>
<td>0.33</td>
</tr>
</tbody>
</table>

tile frequency: the number of times that a tile is watched in the center of user’s FoV, measured with all users on the same video.
Result: tile interval distribution

Question to answer: how long will the user’s gaze stay on the same tile?

Purpose: to predict tile transition

Method: normalize the stay time and find the most suitable distribution
Summary about user behavior analysis

(1) User behavior changes **randomly** along the time;

(2) only **half of tiles** will be viewed (at the center of eyes focus);

(3) **Beta distribution** could be used to simulate the tile interval

(4) the behavior of different types of video follows **the same distribution**, but with some variation;

More can be done, please refer to Dongbiao He, Cédric Westphal et al., IFIP Networking 2019 for that
Outline

• Background & Challenges
• User Behavior Analysis
• Edge Caching & Prefetching
• Evaluation
• Concluding Remarks
Video as a unit for popularity estimation

Possible benefits:

- **Easy to implement**, e.g., reuse existing algorithms
- **Reduce jitter** in video quality

**Method:**

**the self-exciting point process** to reserve the benefit of both LRU and LFU

\[
\lambda \sum_{t'=1}^{t} n_i(t') \phi(t - t')
\]

“**Frequency**” (Kernel Function)
Tile requirement estimation

Tile as a unit for caching

**Method:** Static Analysis (for caching in advance) + Dynamic Analysis (for prefetching)
Region Of Interest/ROI

**Philosophy:** most users focus on some specific regions of the picture => ROI

**Param.:** #ROIs & Distance between ROIs
Dynamic analysis

To **prefetch missing tiles** based on
- locality of user movement
- RTT

<table>
<thead>
<tr>
<th></th>
<th>100ms</th>
<th>250ms</th>
<th>500ms</th>
<th>750ms</th>
<th>1000ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>95%</td>
<td>0.147</td>
<td>0.433</td>
<td>3.012</td>
<td>3.093</td>
<td>3.107</td>
</tr>
<tr>
<td>90%</td>
<td>0.096</td>
<td>0.255</td>
<td>0.567</td>
<td>1.11</td>
<td>2.983</td>
</tr>
<tr>
<td>85%</td>
<td>0.073</td>
<td>0.19</td>
<td>0.401</td>
<td>0.645</td>
<td>0.956</td>
</tr>
</tbody>
</table>

Viewport variation in duration $\theta$

Distance $d$

Refer to Dongbiao He, Jinlei Jiang et al., ICWS 2021 for more details
Bitrate determination for video caching

Challenge: shared bandwidth

Reactive Caching:
➢ Choose the video resolution based on “the average available bandwidth over a period”

$$\delta(t) = \frac{|Bw_t - Bw_{t-1}|}{\min\{Bw_t, Bw_{t-1}\}}$$

Proactive Prefetching:
➢ Triggered after a cache miss happens
➢ Predict and prefetch tiles (identified by id) to be accessed soon but not in the cache yet
➢ Adapt to the real-time end-to-end delay

$$bitrates(\tau) = \frac{Bw \times \tau \cdot t}{|d| \times |\pi|}$$
CUBIST implementation

- The storage would be the bottleneck

<table>
<thead>
<tr>
<th>Device</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDD</td>
<td>36.8 MBps</td>
</tr>
<tr>
<td>SSD</td>
<td>765 MBps</td>
</tr>
<tr>
<td>DRAM</td>
<td>48 GBps</td>
</tr>
<tr>
<td>5G</td>
<td>Max 10 Gbps</td>
</tr>
</tbody>
</table>

- Place items with the caching reward

\[ G_i(\tau) = r_i \sum_{\tau \in i} (\text{size}(\tau) \times \tau.f[T(u, s_i) - T(u, \text{cache})]) \]

- Hierarchical cache

- \( r_i \): the popularity of the video
- \( T(u, s_i) \) and \( T(u, \text{cache}) \): the cost to get the segment
- \( \tau.f \): the ratio \( \tau \) is accessed;

- L1 has enough space: assign \( \tau \) a lifetime and put it into L1;
- Move the last ranked tiles to L2
Outline

• Background & Challenges
• User Behavior Analysis
• Edge Caching & Prefetching
• Evaluation
• Concluding Remarks
CUBIST evaluation: settings

• Datasets
  • 25 videos
  • 109 users

• Requests & Bandwidth
  • User requests: GlobeTraff
  • Bandwidth variation: 4G Trace

• Benchmarks
  • Video Cache, CUBIST-NP
  • Tile Cache

  • X. Corbillon, F. De Simone, and G. Simon. 360-degree video head movement dataset.

  • J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck. HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks.

  • A. Mahzari, A. T. Nasrabadi, A. Samiei, and R. Prakash. Fov-aware edge caching for adaptive 360° video streaming. MM 2018
Evaluation: benefit of hierarchical cache

1) the cache space is 20% of the total video size
2) the ratio of L1 to L2 cache is 3:2
3) the ratio of L1 to L2 hit varies between 9:1 and 7:3 randomly

CUBIST improves the throughput from 765MBps to 39GBps

Highlight: hierarchical cache design with fixed cache cost means larger cache space and higher cache hit ratio, which would bring in more benefit.
Evaluation: benefit of prefetching

- Video quality and cache hit ratio are balanced
- Utilize network resources better
Evaluation: benefit of caching

- Tile Prefetching gets **10%** more gains for caching.
- CUBIST costs **20%** less caching space than Tile Cache.
Evaluation: QoE of videos

✓ Compared with Tile Cache, CUBIST only needs half of the video transitions
✓ CUBIST outperforms Tile Cache, whose median bitrate is 26.9Mbps, by 12.9% in video bitrate
Outline

• Background & Challenges
• User Behavior Analysis
• Edge Caching & Prefetching
• Evaluation
• Concluding Remarks
Conclusions

Immersive video streaming is challenging
• Ultrahigh bandwidth requirement
• Ultralarge Storage Requirements
• Ultralow Motion-to-Photon Delay

CUBIST employs edge caching to solve the problem
• Video-based popularity estimation → simplified implementation
• Proactive tile prefetching → more cache hit
• Hierarchical cache organization → reduced cache node cost
• Bitrate determination: Clients <-> Edge Nodes <-> Servers → better QoE
Limitations and future work

Limitations

• Not applicable to live immersive video streaming
• No consideration of joint caching at multiple edge servers

Future work

• More effective algorithms for tile caching and prefetching, possibly via machine learning
• Coordinated caching at multiple edge servers
• More efficient video coding scheme for transmission
• In-network quality enhancement or even tile generation
Thanks for your attention!