Had You Looked Where I'm Looking? Cross-user Similarities in Viewing Behavior for 360° Video and Caching Implications

Niklas Carlsson
Linköping University, Sweden

Derek Eager
University of Saskatchewan, Canada

ABSTRACT

The demand and usage of 360° video services are expected to increase. However, despite these services being highly bandwidth intensive, not much is known about the potential value that basic bandwidth saving techniques such as server or edge-network on-demand caching (e.g., in a CDN) could have when used for delivery of such services. This problem is both important and complicated as client-side solutions have been developed that split the full 360° view into multiple tiles, and adapt the quality of the downloaded tiles based on the user's expected viewing direction and bandwidth conditions. To better understand the potential bandwidth savings that caching-based techniques may offer for this context, this paper presents the first characterization of the similarities in the viewing directions of users watching the same 360° video, the overlap in viewports of these users (the area of the full 360° view they actually see), and the potential cache hit rates for different video categories and network conditions. The results provide substantial insight into the conditions under which overlap can be considerable and caching effective, and can inform the design of new caching system policies tailored for 360° video.

CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Networks → Network performance evaluation**.

KEYWORDS

360° streaming, caching, viewport overlap

ACM Reference Format:
Niklas Carlsson and Derek Eager. 2020. Had You Looked Where I'm Looking? Cross-user Similarities in Viewing Behavior for 360° Video and Caching Implications. In *Proceedings of the 2020 ACM/SPEC International Conference on Performance Engineering (ICPE '20), April 20–24, 2020, Edmonton, AB, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3358960.3379129

1 INTRODUCTION

Interactive streaming [1, 5, 10, 12, 21, 29, 34] such as 360° video put the users in control of their viewing direction and have the opportunity to revolutionize what users expect from their viewing experiences. Already today, popular services such as Facebook and YouTube offer large catalogues of 360° content. With rapidly increasing 360° content catalogues and the introduction of relatively inexpensive user interfaces (ranging from smartphone-based solutions to dedicated head mounted displays), the demand for 360° streaming services can only be expected to increase.

With 360° streaming services being highly bandwidth intensive, identifying and understanding bandwidth saving opportunities in the wide-area delivery of 360° video is therefore becoming an increasingly important problem. Perhaps the most popular bandwidth saving opportunity studied in the research literature is based on the observation that, with 360° video, only a limited fraction of the full view (called the *viewport*) is displayed at each point in time. Motivated by this observation, to reduce the bandwidth usage and to improve the expected playback quality given a fixed bandwidth, different streaming delivery techniques have been studied that allow alternative playback qualities to be delivered for each candidate viewing direction [2, 6, 12, 20, 29, 30].

With video delivery systems using HTTP-based Adaptive Streaming (HAS), a video is split into chunks (e.g., 2-5 seconds in duration) that are each encoded at multiple quality levels, allowing clients to adapt their playback quality based on current network conditions, for example, and to build up a buffer to protect against stalls that may be caused by future bandwidth variations. With 360° video, each chunk can further be split into multiple tiles, each corresponding to a portion of the 360° view. This division into tiles complicates prefetching, since now, when prefetching data from a future chunk, the client player needs to determine which tiles from the chunk to prefetch and a quality level for each. The prefetching policy must address a prefetch-aggressiveness tradeoff [1] and balance the use of a larger buffer (to protect against stalls) against making prefetching decisions closer to the time of playback (improving predictions of future viewing directions). To address this problem various head-movement prediction techniques have been proposed and evaluated [2, 26, 35]. However, prior work has not considered the implications of tiling and associated quality-adaptive prefetching techniques for 360° video on the performance of content caches.

Content caches will be more effective the greater the overlaps in the data requested by clients. In this paper we carry out the first analysis, to the best of our knowledge, on the similarities in the viewing directions and viewports (i.e., the area of the full 360° view that each user sees) of users watching the same 360° video, and then analyze and discuss the implications these findings may have on caching performance.

The paper has three main parts, with the second and third parts building on the prior parts. First, we present a general analysis of the similarities in viewing direction among different users when at identical playback points within the same 360° video; e.g., as measured by angular differences of the viewing directions, overlap in viewports, and how the viewport's overlap with the aggregate

ICPE '20, April 20–24, 2020, Edmonton, AB, Canada
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6991-6/20/04…$15.00
https://doi.org/10.1145/3358960.3379129

view cover from prior user views increases with the number of such users. This analysis provides insight into inherent similarities in viewing behavior, using measures that are not affected by the details of how video data is delivered to users.

Second, we extend the analysis to evaluate the impact of chunk granularity on the insights from the first part of the paper. This analysis is important to understand how similarities in viewing direction would impact caching performance. For example, consider the case where two users have significantly different viewing directions at a particular time instant, and yet over the time duration of a chunk have essentially the same viewport cover.

Finally, we present a trace-based analysis in which we simulate a proxy cache and evaluate the cache hit rates observed when using prefetching algorithms that attempt to adaptively select which tiles to download and the quality of each to optimize the user's quality of experience. The observed cache hit rates reflect not only viewing direction similarities, both at identical time instants (as analyzed in the first part of the paper) and over the time duration of a chunk (as analyzed in the second part of the paper), but also differences in chunk quality selections caused by bandwidth variations and uncertainties in viewing direction prediction.

Throughout our analysis we use head-movement traces collected for different categories of 360° video [1], allowing us to provide insights into how viewing direction similarities and potential caching performance depend on the nature of the 360° video content. For evaluation of cache performance, we combine our traces with previously collected network measurements capturing a wide range of network conditions [18, 28].

The results provide substantial insight into the conditions under which overlap can be considerable and caching effective. Particularly noteworthy perhaps are the substantial differences observed between different video categories and, in some cases, playback positions within the video. For example, the category of videos for which "the main focus of attention is deemed to always be at the same location in the video" [1] appears to provide the greatest opportunities, among the categories we consider. However, this is not the case until 20-30 seconds into these videos, as viewers of these videos often have an initial exploratory phase during which viewing similarities are smaller compared to the "moving focus" category of videos or the "rides" category.

Our measurement and analysis results can inform the design of new caching system policies tailored for 360° video. Our results may also have implications for other policies. For example, cache hit rate may benefit from cap-based network solutions that stabilize the bandwidth seen by individual clients (e.g., [14]). With respect to prefetching policies, our results show how the value of using the viewing directions of previous users for viewport prediction varies among different 360° video categories and, in some cases, also depends on the playback position.

The remainder of the paper is organized as follows. Section 2 presents background and introduces the head-movement dataset used here. Section 3 presents our analysis of viewing direction similarities between pairs of users at identical playback points, pairwise viewport overlaps, and viewport overlaps with aggregate view covers from different numbers of prior users. Section 4 extends this analysis to take into account the chunk granularities used, before Section 5 presents our trace-based simulations of cache performance under different network bandwidth conditions and uncertainties in viewing direction prediction accuracies. Finally, Section 6 presents related work and Section 7 concludes the paper.

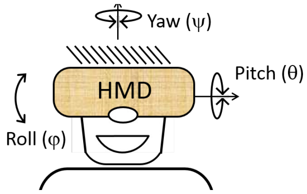


Figure 1: Head-movement coordinates: Yaw, pitch, and roll.

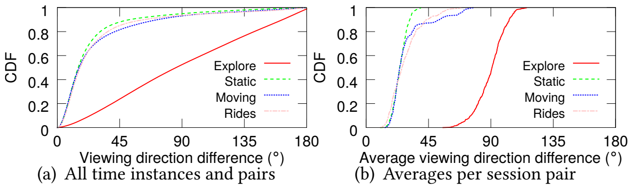
2 BACKGROUND AND DATASET

360° videos capture the view in all directions and allow users to look in any direction at each point during playback; e.g., by moving their head while wearing a head mounted display (HMD). While 360° videos also can be viewed in the browser on PCs, on smartphones, or on tablets, for the work presented here we assume use of an HMD. Figure 1 shows an example user wearing an HMD and defines the viewing angles (i.e., yaw, pitch, and roll) used in our work.

All angles are measured in degrees and normalized so that two users will have the same recorded viewing direction at a given point during their viewing of the same video whenever their viewports completely overlap, regardless of original head positioning. Here, yaw (±180°) measures sideways rotations (relative to a 0° line corresponding to the initial viewing direction as set in the video), the pitch (±90°) vertical head rotations (relative to a horizontal plane), and the roll (±90°) rotations of the head (relative to holding the head straight).

For our analysis we use a dataset collected by Almquist et al. [1]. The dataset consists of fine grained head-movement data collected when 32 people watched 360° videos from a set of 30 such videos. The videos were downloaded and played in 4K resolution, were 1-5 minutes long (3 min. on average), and were (by the authors) split across five categories [1, p. 260]: *exploration* ("no particular object or direction of special interest and the users are expected to explore the entire sphere throughout the video duration"), *static focus* ("the main focus of attention is deemed to always be at the same location in the video"), *moving focus* ("story-driven videos where there is an object of special interest that is moving across the 360° sphere"), *rides* ("the users take a virtual ride in which the camera is moving forward at a high speed, making users feel that they too are moving forward quickly"), and *miscellaneous* ("includes videos that were deemed to have a mix of the characteristics of the other categories or had a hard-to-classify, unique feel, to them").

In total, the dataset includes head movements from 439 unique viewings (totaling 21 hrs and 40 min). The "semi-random" design of the user study ensured that all 32 users watched one "representative" video from each category, while the other videos got between 8-13 views each. In this paper, we focus on the representative videos for the first four (more well-defined) categories, as these allow for a richer head-to-head comparison of the similarities and differences in viewing direction, and hence also of the caching opportunities, when multiple viewers watch the same video. Additional results are presented in [4]. Since Almquist et al. found that yaw movements dominate, followed by pitch, with only small roll movements, we focus only on yaw and pitch.



(a) All time instances and pairs (b) Averages per session pair

Figure 2: CDFs of pairwise viewing direction differences; representative videos.

3 SIMILARITY CHARACTERIZATION

In this section we present an initial characterization of the viewing similarities and differences between users watching the same video. For each video, we calculate and report summary statistics based on the viewing directions observed every 50ms. To account for the timestamps not always aligning perfectly between the traces, we use interpolation and note that good accuracy is ensured by the use of a measurement granularity of 10ms (i.e., 100 measurements per second) in the data collection.

3.1 Pairwise viewing differences

First, we consider the difference in viewing direction of two users at identical playback points within the same video, as measured by the angle between these directions. Figure 2 shows cumulative distribution functions (CDFs) of the pairwise differences, when combining the differences in both yaw and pitch, for all pairs of viewing sessions of each of the representative videos. (For each of these videos we have 32 user traces and therefore 496 pairs.) In particular, Figure 2(a) shows CDFs for the differences, as measured every 50 ms throughout the viewing sessions, and Figure 2(b) shows CDFs for the average of these differences for each session pair.

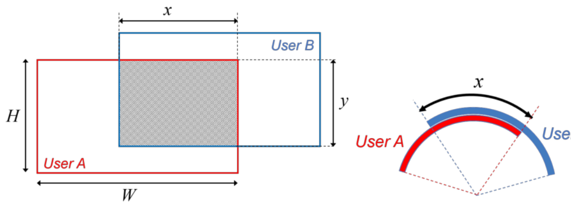
As expected, the pairwise differences are substantially larger for the *explore* category than for the other categories (i.e., *static*, *moving*, and *rides*). For example, the close-to-straight *explore* line in Figure 2(a) suggests that the viewing directions of users watching *explore* videos are close to independent. In contrast, for the other categories the view angle differences are less than 45° for 80% of the time instances, showing that viewers of these three video categories often are looking at the same parts of the video.

These significant differences among the categories are also clearly visible when considering the viewing direction difference averaged over the entire playback duration (Figure 2(b)) and when considering the average differences also for the other videos in the dataset.

While the above results are based on the total directional differences across both yaw and pitch, the observations (and values) are very similar when focusing on yaw only.

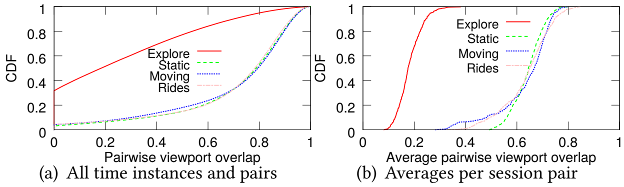
3.2 Pairwise viewport overlap

Similarities in what content clients download and what they actually watch depend not only on the users' viewing directions, but more importantly on their viewports. Here, we consider two types of viewports. First, we consider the 2D area of the viewing field being displayed. Second, motivated by most head movements being along the yaw angle, we consider a *sliced version*, in which we ignore the pitch and only consider the yaw angle. In both cases, we report overlaps normalized by the total viewport size. Figure 3 illustrates the metrics.



(a) Yaw + pitch (b) Yaw only

Figure 3: Definition of the pairwise viewport overlap metric. (Handling of wraparound effects are described in [4].)



(a) All time instances and pairs (b) Averages per session pair

Figure 4: CDF normalized pairwise viewport overlap.

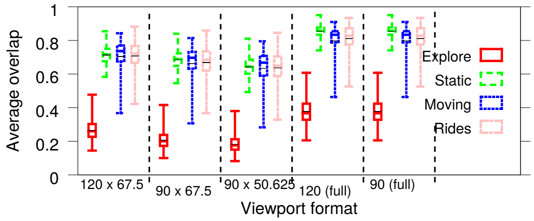
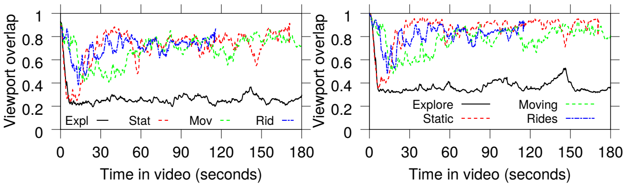


Figure 5: Average normalized pairwise overlap for representative videos, when using different viewports.

Figure 4 shows CDFs of the normalized pairwise overlap for the representative videos when using a 120×67.5 viewport. Here, Figure 4(a) shows CDFs for the pairwise overlap at identical playback points (as measured every 50 ms) and Figure 4(b) shows CDFs for the average of these overlaps for each session pair. As before we observe significant differences when comparing the *explore* category with the other categories. For example, with the *explore* video, more than 35% of the time there is no pairwise overlap, whereas for the other categories there is at least a 50% overlap in more than 80% of the instances. (See Figure 4(a).) Considering the average normalized pairwise viewport overlap (Figure 4(b)), no pair of *explore* sessions had an average overlap of more than 40%, while less than 6.5% of the *moving* session pairs, less than 1% of the *rides* session pairs, and none of the *static* session pairs had an average overlap that did not exceed 40%. In fact, for these three categories, more than 70% of the sessions see an average overlap of at least 60%.

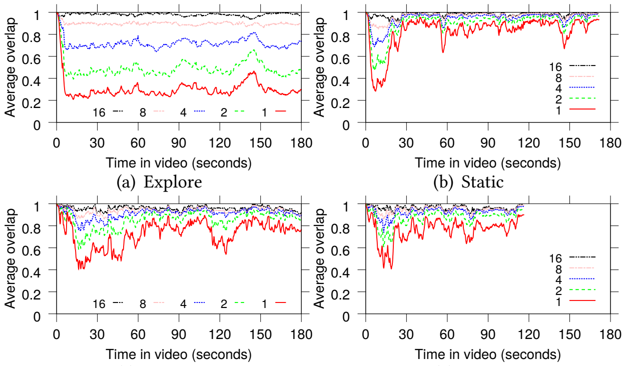
Impact of viewport: Figure 5 shows summary statistics for five alternative viewports (the last two ignoring differences in pitch). For each class we show the minimum over all pairs of sessions of the average viewing direction difference (bottom marker), 25-percentile (bottom of box), median (middle colored marker), 75-percentile (top of box), maximum (top marker), and average (black marker). We note that as the viewports become larger, the overlaps increase.

Longitudinal playpoint dependencies: Note that pairwise overlaps vary over the playback duration. For example, all clients start with the same viewing direction and prior work [1] has shown that with *static* videos there is often an initial exploration phase. Figures 6(a) and 6(b) show the overlap averaged over all session pairs as a function of the time from the start of the video, for two



(a) 120×67.5 (b) 90 full

Figure 6: Time-line plot of the normalized pairwise overlap.



(a) Explore (b) Static

(c) Moving (d) Rides

Figure 7: Average normalized viewport overlap as function of time. (Viewport size $W = 90$.)

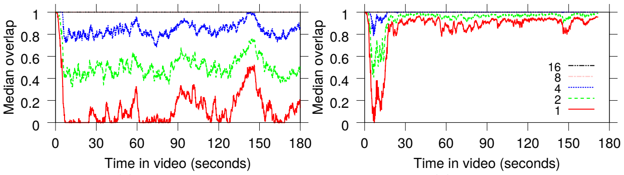
example viewports; the first (120×67.5) taking pitch into consideration and the second (90 full) ignoring pitch. In addition to smaller initial average overlaps for the *static* video, resulting from initial exploration, we also observe a somewhat smaller average overlap at the beginning of the videos in the *rides* and *moving* categories than towards the end of those videos. This suggests that cache hit rates may improve over the duration of many video sessions.

3.3 Viewing sequence analysis

To gain insight into how the potential cache performance may be impacted by the number of users having watched a video, we now look beyond pairwise viewport overlaps and consider overlaps among larger sets of users. Specifically, we evaluate how the viewport overlap with the aggregate view cover from prior user views increases with the number N of such users.

For each representative video, we created 1,000 random orderings of the 32 viewing sessions recorded in the dataset for that video, and for each sequence and viewing session, evaluated the overlap at identical playback points between the respective user's viewport and the aggregate viewing area covered by all prior users in that viewing sequence. For this analysis, we ignore pitch (i.e., use vertically sliced viewports) and for each time instance and session sequence, we first merge the viewport coverage of all N prior users into a number of non-overlapping (merged) viewport areas. Then, we calculate the overlap of these non-overlapping (merged) viewport intervals with the current user's viewport, before adding this user's viewport to the merged intervals and repeating the calculations for the next user in the sequence.

Figure 7 shows timeline plots of the average (over the 1,000 random orderings of viewing sessions) normalized viewport overlap. Note that the benefits of more prior video viewings increase when there is more variability in where users are looking (e.g., *explore* videos or the beginning of the *static* video). Interestingly, the improvements are even larger for median overlap, as seen in Figure 8. The larger median improvements show that the majority of the sessions quickly see significant benefits from each additional prior client. For example, with just four prior clients, in the case of the *static* video, the majority of clients have 100% overlap from roughly the 15 second mark.



(a) Explore (b) Static

Figure 8: Median normalized viewport overlap as function of time. (Viewport size $W = 90$.)

4 CHUNK GRANULARITY ANALYSIS

It is important to remember that caching (and video delivery itself) typically is done on a per-chunk basis. The viewport may change during the playback duration of a chunk, resulting in a larger per-chunk viewport cover than the viewport at an individual playback point. The overlaps between per-chunk viewport covers and those of prior clients are important in caching. We next study and report on per-chunk statistics.

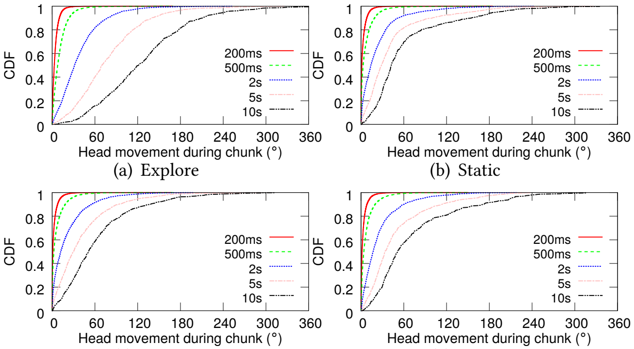
4.1 Changes in viewing direction

Figure 9 shows CDFs of a bound on the maximum viewing direction change over a chunk duration, as calculated using the fine-grained measurements in our dataset [4], for the representative videos and a range of chunk durations (200ms-10s). As before, the *explore* category stands out, with much larger head movements. However, note that for intermediate chunk durations (e.g., 2s), the head movements still only cover a small fraction of the view field. For example, for the representative videos the maximum viewing direction changes for 80% of the chunks are upper bounded by 57.7°, 34.5°, 36.3°, and 38.7°, respectively.

4.2 Per-chunk viewport cover

To measure the total viewing area that is included within a user's viewport for at least some portion of a chunk's playback period, we calculate a bounding box of the head movements during this time period, which we term the *per-chunk viewport cover.* In the following, we report per-chunk viewport covers normalized by the total size of the viewport.

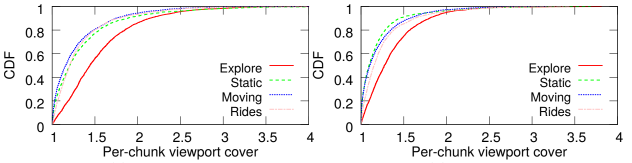
Figure 10 presents CDFs of the normalized per-chunk viewport cover size for each of the four representative videos, for 2 second chunks, and two viewport sizes (120×67.5 and 90 full) with maximum theoretic normalized cover sizes of 8 and 4, respectively. Note that the coverage is typically much smaller than the theoretic maximum. For example, with the sliced 90 full viewport, 80% of the chunks have a normalized cover size of at most 1.57, 1.28, 1.35, and 1.38, respectively, for the four representative videos. These small cover sizes suggest that tiles could fruitfully be prioritized on a per-chunk basis since a significant portion of the potential viewing area is not viewed during the playback of a chunk.



(a) Explore (b) Static

(c) Moving (d) Rides

Figure 9: Impact of chunk duration on the change in viewing angle for the representative videos.



(a) 120×67.5 (b) 90 full

Figure 10: Normalized per-chunk viewport cover size. (2s chunks.)

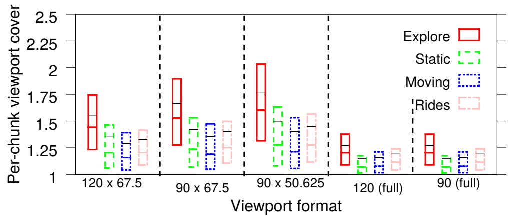


Figure 11: Impact of viewport format on the normalized per-chunk viewport cover size. (Chunk duration of 2s.)

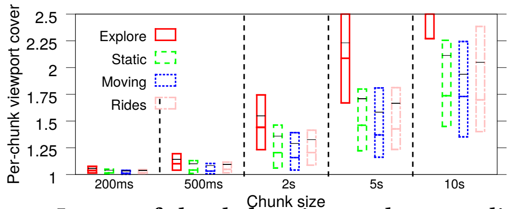
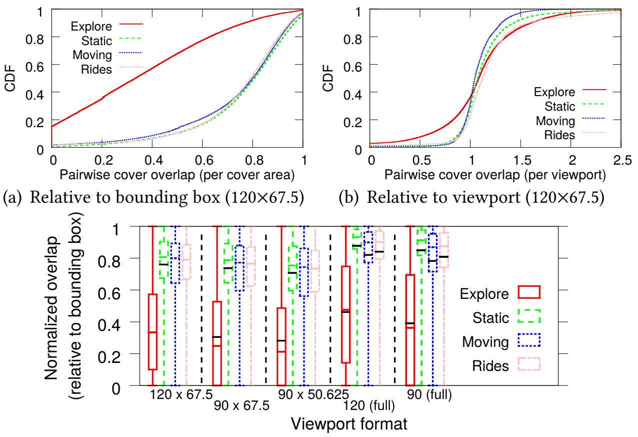


Figure 12: Impact of chunk duration on the normalized per-chunk viewport cover size. (Viewport size 120×67.5.)

Figure 12 shows the impact of the chunk duration on the normalized per-chunk viewport cover size for the representative videos. Focusing on the 75-percentile values, except for the cases of (i) *explore* using a chunk duration of 5 or more seconds, and (ii) extremely long duration chunks of 10 seconds, the normalized per-chunk viewport cover size is again consistently below 2. In general, for *static*, *moving*, and *rides* this cover size is substantially smaller.

4.3 Pairwise cover overlap

We next combine our techniques for analysis of pairwise viewport overlap and for determining per-chunk viewport covers, to measure the pairwise overlap in per-chunk viewport cover.



(a) Relative to bounding box (120×67.5) (b) Relative to viewport (120×67.5)

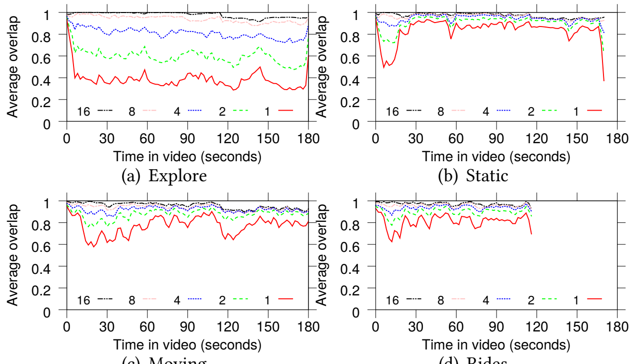
(c) Different viewports

Figure 13: Pairwise overlap in per-chunk viewport cover.

The first two sub-figures in Figure 13 show the pairwise overlap in per-chunk viewport cover normalized relative to the size of the user's bounding box and relative to the viewport size, respectively, when using a 120×67.5 viewport, and the third sub-figure shows summary statistics (of the first kind) also for other viewports. We note that the *explore* category stands out even more than we have seen before, when considering the overlap normalized relative to the bounding box size (Figures 13(a) and 13(c)). For example, referring to Figure 13(a), while there is at least a 60% overlap in cover for 79-83% of the chunks for the *static*, *moving*, and *rides* videos (83%, 79%, and 83%, respectively), the corresponding fraction of chunks is only 23% for the *explore* video. This reflects the fact that the videos in the *explore* category typically have both larger head movements during a chunk duration, and larger pairwise viewing direction differences (including during the chunk playback period). Furthermore, the variations in the absolute overlap (e.g., as normalized relative to the viewport size, as in Figure 13(b)) are much greater for the *explore* video, and conversely, the variations are smallest for the *static* video.

4.4 Request sequence analysis

We next extend our analysis of the overlap with the aggregate view cover from *N* prior user viewings of the same video to account for chunk boundaries. Throughout this section we use 2 second chunks, a sliced 90° viewport, and normalize the reported cover overlap relative to the size of the cover of the user of consideration. Figure 14 shows the overlaps across the playback durations of the example videos. Note the larger overlaps compared to those in Figure 7, although the qualitative differences among the results for the representative videos are quite similar. In fact, during the first 120 seconds of the *explore* video and the initial explore phase of the *static* video the average overlaps when there are *N*=8 and *N*=16 prior users are close to one. In general, however, the overlaps when there are fewer prior users (e.g., *N*=1, *N*=2, and *N*=4 curves) are greater when users are less exploratory (e.g., with *moving*, *rides*, and after the initial exploratory phase of the *static* video). These chunk-level results again highlight important differences in the caching opportunities that different video categories present, and that videos of some categories (e.g., *static*) may require different



(a) Explore

(b) Static

(c) Moving

(d) Rides

Figure 14: Chunk-based time plot of the average normalized cover overlap. (Viewport size $W = 90$.)

optimizations for the initial (exploratory) phase than the later parts of the videos.

5 CACHE PERFORMANCE SIMULATIONS

Our trace-driven cache simulations take into account multiple sources of uncertainty that impact prefetching and caching performance. First, network bandwidth varies over time and clients do not know their future bandwidth. Second, with different scenes requiring different encodings, for example, chunk sizes typically vary from chunk-to-chunk and across different parts of the same video. These first two uncertainties result in variable download times and buffer sizes, as clients adapt the requested encodings so as to try to maintain a relatively stable buffer and to avoid stalls. Third, as seen here, with 360° video there is a lot of variability and uncertainty in how users move their heads. The client player can try to predict head movements and prefetch high quality tiles only for some directions, but prediction accuracy will vary across videos as well as over time during video playback.

5.1 Simulation model

To better understand the impact that download speed variability (caused by the first two uncertainties) and the view direction prediction accuracy have on the cache efficiency under different quality selection algorithms, we use a simple simulation model where we use probability distributions to capture each of the uncertainties.

In our model [4], we assume that client player i makes its tile selection for each chunk k based on a quality of experience (QoE) optimization problem taking into account (i) the capacity $C_{i,k}$ drawn from a distribution $P_C(C)$, and (ii) the probability $P_n(n)$ that a specific tile n will be viewed (where the probability $P_n(n)$ depends on the class of videos considered and how far in advance of playback the client must make its tile selection for the chunk).

Finally, to account for the third uncertainty, the predicted viewing direction used when solving the optimization is offset from the actual viewing direction at playback time by an angle $\psi^{\epsilon}_{i,k}$ chosen by sampling from a probability distribution $P_{\psi}(\psi)$.

To obtain a hit rate estimate for a particular video and number of previous clients, we average results from 1,000 simulations, each with 32 randomly-ordered users sequentially viewing the video. Each client uses the user head movements recorded in our trace dataset for that user when viewing the respective video. We assume that the system always starts with an empty cache and measure how the hit rate (both in terms of tile objects and bytes delivered) changes as more and more users view the same video.

5.2 Parameters and example distributions

For the distribution $P_C(C)$ we use distributions obtained by drawing random samples from two real-world datasets, and two synthetic distributions. The real-world datasets are: (i) 10,000 download bandwidth measurements collected by mobile 3G and 4G users of a dominant national speed testing service [18] over a 19 hour window on Feb. 15-16, 2015, and (ii) 10,000 sample points from "bus" commuter traces collected in Norway by Riiser et al. [28] between Aug. 28, 2010, and Jan. 31, 2011. The synthetic distributions we use are: (i) a distribution in which the bandwidth capacity C varies across three different levels such that C is equal to the average bandwidth 40% of the time, twice the average bandwidth 20% of the time, and half the average bandwidth 40% of the time, and (ii) a constant bandwidth capacity. To account for the fact that bandwidths have increased substantially since the traces in the real-world datasets were collected (2010-2011 and 2015, respectively) and to ensure a more fair head-to-head comparison across the different distributions, we scale the bandwidths in the real-world datasets and choose parameters for the synthetic distributions so that the average bandwidth in each case is the same. Furthermore, we use normalized units so that a normalized bandwidth of 1 corresponds to the bandwidth needed to deliver all tiles at the maximum quality.

To determine choices for the $P_{\psi}(\psi)$ and $P_n(n)$ distributions, we used the yaw angle changes in the traces from the head-movement dataset over different time intervals and for different video categories [4].

Finally, for the example simulation results presented here, we consider a sliced 360° video with each 2-second chunk split into 6 tiles, each covering 60 degrees, and for which the tile encoding rates are each proportional to one of seven quality levels (found in an example YouTube video): 0, 144, 268, 625, 1124, 2217, 4198 (in normalized units).

5.3 Example results

Figure 15 shows a baseline comparison of the tile object hit rates for the representative videos. This figure clearly illustrates that better cache performance is achieved with the *static*, *rides* and *moving* videos compared to with the *explore* video. This observation is not surprising given the results reported in previous sections, and is also consistently seen with other distribution and parameter settings. For example, with four prior clients (i.e., $N = 4$), the object hit rate for the *static*, *rides*, and *moving* videos ranges between 0.75-0.80, while for the *explore* video it is only 0.64. Note also that these differences can have a large impact on bandwidth requirements and cache write costs, for example, as the object miss rate for *explore* is 80% higher (a factor of 0.36/0.20) than for the *static* video. In the reminder of this section we present results only for the two extreme cases of *static* and *explore* videos, but note that the results for *rides* and *moving* are relatively similar to those of *static*.

Object vs byte hit rates: Figure 16 compares the byte and object hit rates for the *static* and *explore* videos. The higher byte hit rates suggest even better cache benefits than suggested by the object

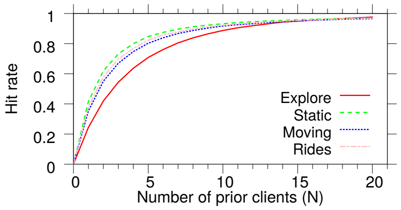


Figure 15: Object hit rate for trace-based simulations with fixed bandwidth.

hit rate results. The observed differences in byte hit rate (between classes) can have a large impact on bandwidth requirements. For example, with four prior clients, the byte hit rate for the *explore* video is 0.73 while that for the *static* video is 0.85, implying an 80% higher byte miss rate for the *explore* video.

Impact of client's bandwidth variability: As seen in Figure 17, hit rates typically reduce the greater the bandwidth variability. Note however that the relative impact of bandwidth variability is smaller for the *explore* video than the *static* video, showing that higher uncertainty in viewing direction and bandwidth do not contribute independently to reduced hit rates.

Impact of average bandwidth: In Figure 18 we show additional object hit rate results for the two extreme cases of (a) constant bandwidth (identical for all clients) and (b) the bandwidth distribution obtained by drawing random samples from measurements collected by mobile 3G and 4G users of a dominant national speed testing service [18]. In practice, we expect clients sharing the same cache to see bandwidth variation between these two extremes, with operators likely to strive towards providing increasingly stable network conditions for streaming clients [14]. When interpreting these results, it is important to note that clients sharing an edge-cache (e.g., operated by a CDN or in cooperation with a CDN) might be expected to experience more similar bandwidth conditions than in the speed testing data. Also, with the introduction of cap-based solutions [14], and other streaming-aware network solutions, used by different operators to stabilize HAS performance, improve QoE, and to reduce unnecessary bandwidth usage, it seems likely that many networks in the future will provide fairly stable conditions for their streaming clients. Therefore, we believe that likely bandwidth variations fall between these two extremes.

Figure 18 shows that our default case of a normalized bandwidth of 0.476 results in close to the worst-case hit rates, suggesting that the hit rates with tiled 360° video could be greater than suggested by Figure 15. Also, when comparing Figures 18(a) and 18(b) it should be noted that owing to our choice of normalized units for bandwidth, the hit rate is always one when all clients have the same (constant) bandwidth above one (Figure 18(a)), whereas bandwidth variations in the national speedtest dataset result in significant periods of bandwidth below one even for average values substantially larger than one.

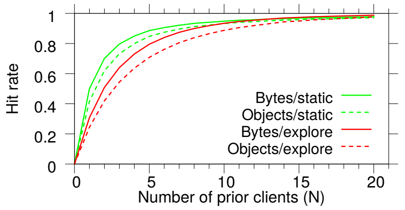
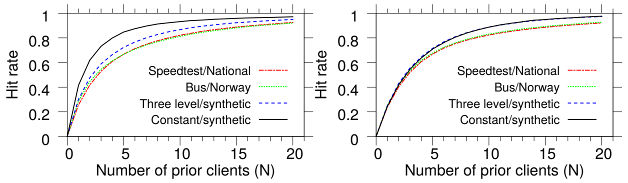


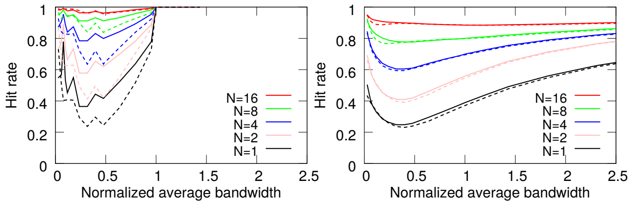
Figure 16: Object hit rate vs byte hit rate for trace-based simulations with fixed bandwidth.



(a) Static

(b) Explore

Figure 17: Object hit rate for trace-based simulations using different network bandwidth profiles.



(a) All same bandwidth

(b) National speedtest samples

Figure 18: Impact of the normalized average bandwidth: *static* (solid lines) and *explore* (dotted lines).

6 RELATED WORK

Broadly, the related work can be split into works that consider the head movements during viewing of 360° videos, and caching of HAS videos. While some recent works have considered optimized cache management policies for 360° videos [19, 22, 24], none of these works provide a data-driven characterization of the caching opportunities that would be observed with traditional caching policies that simply cache the requested tiles when the clients apply adaptive prefetching techniques.

Head-movement characterization: Some recent works have collected datasets and characterized the 360° viewer behavior [1, 2, 7–9, 20, 25]. However, most of these datasets use relatively short video segments and do not capture changes in behavior over time or across classes of videos. The primary exception, and the work most closely related to ours, is the work by Almquist et al. [1], as we use their dataset. In their work, they present a category-based characterization of the head movements over time, and analyze how changes in viewing behavior depend on the time window considered, but do not consider overlapping viewports of users watching the same video or other similarity metrics of users' viewing directions.

Caching for HAS: Prior works have characterized the caching opportunities for HAS content in mobile networks [11], evaluated the impact that cross traffic has on cache performance [3], identified HAS specific instabilities and other tradeoffs associated with the use of caches combined with HAS [13, 15], and proposed HAS-aware solutions to improve the client performance in such scenarios [13, 15, 17, 23, 31, 32]. Other works have considered various cache management problems in the context of HAS [16, 36] and optimized replication for interactive multiview streaming [27, 33].

Most closely related to our work are perhaps recent works that present optimized cache management solutions for 360° video [19, 22, 24]. These works formulate optimization problems related to the caching of tiled 360° videos [19, 24] or try to learn probabilistic models of users FoV for each video so to improve cache performance [22]. However, none of the papers presents a data-driven characterization of the users' viewport overlaps and the bandwidth saving opportunities this provides basic caching policies. Here, we present the first such data-driven analysis of similarities in head movements between users watching the same video, the users' viewport overlaps, and their implications on caching of tiled 360° videos belonging to different categories.

7 CONCLUSIONS

This paper uses head-movement traces for different categories of 360° videos, including *explore*, *static*, *moving*, *rides*, to characterize similarities in the viewing directions and viewports of users watching the same video, how these example metrics differ between the different video categories, and to analyze and discuss how such similarities and differences impact the effectiveness of caching tiled 360° videos. To the best of our knowledge, this is the first paper to provide such analysis.

Our results consistently highlight substantial differences between different video categories in the pairwise viewport overlaps observed and their impact on the potential bandwidth savings from caching. For example, with the exception of the initial 20-30 second exploration phase of *static* videos, the *static* videos provide the greatest caching opportunities. However, during this initial phase, their pairwise viewport overlaps are almost as small as for the *explore* videos, which have the smallest overlaps among the categories considered here. In contrast, *moving* and *rides* videos have a less pronounced exploration phase, and overall often provide similar caching opportunities and performance as the *static* videos. Our results also show that improved viewport prediction techniques [35] may not only help improve user QoE, through the use of more accurate prefetching, but may also help increase cache hit rates and reduce bandwidth requirements.

More generally, our results can inform the design of new caching system policies tailored for 360° video, and may also have implications for other contexts than caching. For example, our novel category-based characterization clearly highlights that there are substantial differences among the video categories in the value of using the viewing directions of previous users for viewport prediction. The results also clearly show that cache performance, and hence also likely user QoE, benefit from stable network conditions, motivating the use of cap-based network/server-side solutions or less greedy client-side solutions.

REFERENCES

[1] M. Almquist, V. Almquist, V. Krishnamoorthi, N. Carlsson, and D. Eager. 2018. The Prefetch Aggressiveness Tradeoff in 360° Video Streaming. In *Proc. ACM MMSys.*

[2] Y. Bao, H. Wu, T. Zhang, A. Ramli, and X. Liu. 2016. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In *Proc. IEEE Big Data.*

[3] S. Benno, J. O. Esteban, and I. Rimac. 2011. Adaptive streaming: The network HAS to help. *Bell Lab. Tech. J.* 16, 2 (Sept. 2011), 101–114.

[4] N. Carlsson and D. Eager. 2019. Had You Looked Where I'm Looking: Cross-user Similarities in Viewing Behavior for 360° Video and Caching Implications. (2019). arXiv:1906.09779

[5] N. Carlsson, D. Eager, V. Krishnamoorthi, and T. Polishchuk. 2017. Optimized Adaptive Streaming of Multi-video Stream Bundles. *IEEE Transactions on Multimedia* 19 (July 2017), 1637–1653.

[6] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski. 2017. Viewport-adaptive navigable 360-degree video delivery. In *Proc. IEEE ICC.*

[7] X. Corbillon, F. D. Simone, and G. Simon. 2017. 360-Degreee Video Head Movement Dataset. In *Proc. ACM MMSys.*

[8] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet. 2018. A Dataset of Head and Eye Movements for 360°; Videos. In *Proc. ACM MMSys.*

[9] S. Fremerey, A. Singla, K. Meseberg, and A. Raake. 2018. AVtrack360: An Open Dataset and Software Recording People's Head Rotations Watching 360°; Videos on an HMD. In *Proc. ACM MMSys.*

[10] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen. 2016. Tiling in Interactive Panoramic Video: Approaches and Evaluation. *IEEE Trans. om Multimedia* 18, 9 (Sept. 2016).

[11] A. Gouta, D. Hong, A.-M. Kermarrec, and Y. Lelouedec. 2013. HTTP adaptive streaming in mobile networks: Characteristics and caching opportunities. In *Proc. IEEE MASCOTS.*

[12] M. Hosseini and V. Swaminathan. 2016. Adaptive 360 VR video streaming: Divide and conquer. In *Proc. IEEE ISM.*

[13] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri. 2013. Helping Hand or Hidden Hurdle: Proxy-assisted HTTP-based Adaptive Streaming Performance. In *Proc. IEEE MASCOTS.*

[14] V. Krishnamoorthi, N. Carlsson, and E. Halepovic. 2018. Slow but Steady: Cap-based Client-Network Interaction for Improved Streaming Experience. In *Proc. IEEE/ACM IWQoS.*

[15] D. H. Lee, C. Dovrolis, and A. C. Begen. 2014. Caching in HTTP Adaptive Streaming: Friend or Foe?. In *Proc. ACM NOSSDAV.*

[16] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard. 2018. QoE-Driven Mobile Edge Caching Placement for Adaptive Video Streaming. *IEEE Trans. om Multimedia* 20, 4 (Apr. 2018).

[17] K. Liang, J. Hao, R. Zimmermann, and D. K. Y. Yau. 2015. Integrated Prefetching and Caching for Adaptive Video Streaming over HTTP: An Online Approach. In *Proc. ACM MMSys.*

[18] T. Linder, P. Persson, A. Forsberg, J. Danielsson, and N. Carlsson. 2016. On Using Crowd-sourced Network Measurements for Performance Prediction. In *Proc. IEEE/IFIP WONS.*

[19] K. Liu, Y. Liu, J. Liu, A. Argyriou, and Y. Ding. 2019. Joint EPC and RAN caching of tiled VR videos for mobile networks. In *Proc. MMM.*

[20] W. Lo, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu. 2017. 360° Video Viewing Dataset in Head-Mounted Virtual Reality. In *Proc. ACM MMSys.*

[21] R. Ma, T. Maugey, and P. Frossard. 2018. Optimized Data Representation for Interactive Multiview Navigation. *IEEE Trans. om Multimedia* 20, 7 (July 2018).

[22] A. Mahzari, A. Nasrabadi, A. Samiei, and R. Prakash. 2018. FoV-aware edge caching for adaptive 360 video streaming. In *Proc. ACM Multimedia.*

[23] S. K. Mehr, P. Juluri, M. Maddumala, and D. Medhi. 2018. An adaptation aware hybrid client-cache approach for video delivery with dynamic adaptive streaming over HTTP. In *Proc. IEEE/IFIP NOMS.*

[24] G. Papaioannou and I. Koutsopoulos. 2019. Tile-based Caching Optimization for 360 Videos. In *Proc. ACM MobiHoc.*

[25] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan. 2018. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proc. ACM MobiCom.*

[26] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan. 2016. Optimizing 360 video delivery over cellular networks. In *Proc. All Things Cellular Workshop.*

[27] D. Ren, S.-H. G. Chan, G. Cheung, and P. Frossard. 2014. Coding Structure and Replication Optimization for Interactive Multiview Video Streaming. *IEEE Trans. om Multimedia* 16, 7 (Nov. 2014).

[28] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen. 2013. Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications. In *Proc. MMSys.*

[29] J. Son, D. Jang, and E.-S. Ryu. 2018. Implementing 360 Video Tiled Streaming System. In *Proc. ACM MMSys.*

[30] J. Son, D. Jang, and E.-S. Ryu. 2018. Implementing Motion-Constrained Tile and Viewport Extraction for VR Streaming. In *Proc. ACM NOSSDAV.*

[31] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, M.-L. Champel, and O. Oyman. 2016. Applications and deployments of server and network assisted DASH (SAND). In *Proc. IBC.*

[32] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey. 2017. Enhancing MPEG DASH performance via server and network assistance. *SMPTE Motion Imaging Journal* 126 (Jan/Feb. 2017), 22–27. Issue 1.

[33] L. Toni, G. Cheung, and P. Frossard. 2016. In-Network View Synthesis for Interactive Multiview Video Systems. *IEEE Trans. om Multimedia* 18, 5 (May 2016).

[34] L. Toni and P. Frossard. 2017. Optimal Representations for Adaptive Streaming in Interactive Multiview Video Systems. *IEEE Trans. om Multimedia* 19, 12 (Dec. 2017).

[35] L. Xie, X. Zhang, and Z. Guo. 2018. CLS: A Cross-user Learning Based System for Improving QoE in 360-degree Video Adaptive Streaming. In *ACM Multimedia.*

[36] W. Zhang, Y. Wen, Z. Chen, and A. Khisti. 2013. QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks. *IEEE Trans. on Multimedia* 15, 6 (2013), 1431–1445.