

Technical Disclosure Commons

Defensive Publications Series

March 05, 2018

System for optimizing buffering of 360 degree video streams, by taking current viewing angle into account and buffering more distant viewing angles at separate bandwidths

Anurag Agrawal

Thomas Price

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Agrawal, Anurag and Price, Thomas, "System for optimizing buffering of 360 degree video streams, by taking current viewing angle into account and buffering more distant viewing angles at separate bandwidths", Technical Disclosure Commons, (March 05, 2018) https://www.tdcommons.org/dpubs_series/1081



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

SYSTEM FOR OPTIMIZING BUFFERING OF 360 DEGREE VIDEO STREAMS, BY TAKING CURRENT VIEWING ANGLE INTO ACCOUNT AND BUFFERING MORE DISTANT VIEWING ANGLES AT SEPARATE BANDWIDTHS

Abstract

360-degree video is divided into a plurality of fields of view from a playback device. For example, In the case of four fields of view, referred to as quadrants, the first quadrant is defined as the current field of view of the user, the second quadrant is to the user's right, the third quadrant is directly behind the user, and the fourth is to his left. During playback and buffering, a server decides what video quality (e.g., low/medium/high) to assign to each quadrant of the video based on available bandwidth and the likelihood of a viewer rotating the device to each of the quadrants.

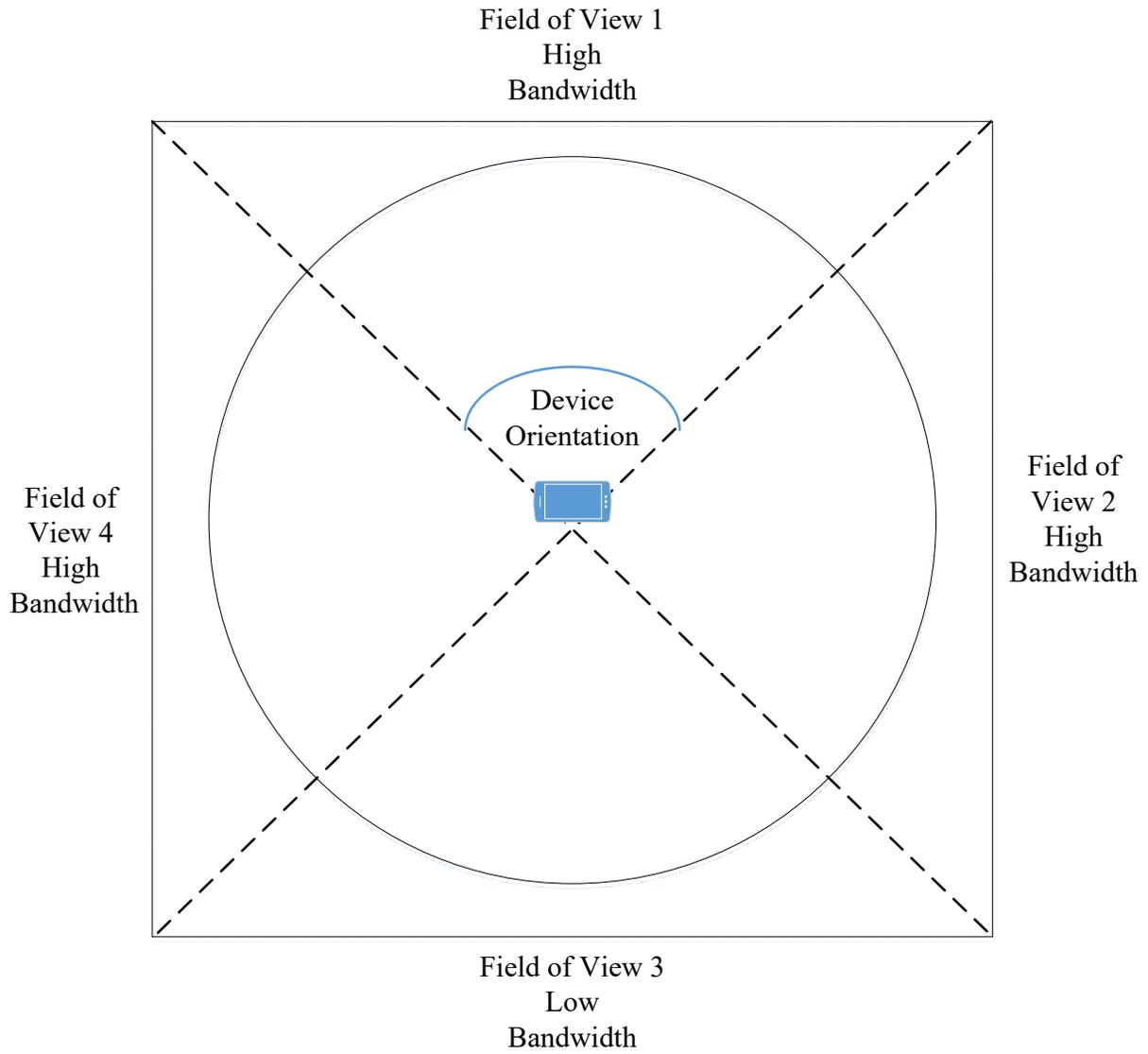
Description

360 degree streaming video is becoming very common; multi-camera rigs that can record in 360 degrees are fairly inexpensive. On the viewer side, users rotate their mobile phone, and accelerometer readings cause the client app that displays the video to show the video angle corresponding to where the user turned. However, as currently implemented, these video streams are often choppy. This is because typically, systems stream all 360 degrees simultaneously, at the same order of priority. Instead, streams should incorporate the viewing angle (or field of view) as a dimension for optimization, based on the assumption of "inertia", that users are less and less likely to turn farther and farther from their current orientation. For example, users may be less than likely to rotate the phone 180 degrees in reverse during video playback. Accordingly, the angles that are farthest away can have their video quality reduced in times when bandwidth is low.

The authors propose splitting 360-degree video into [n] fields of view, each of which is treated differently for the purposes of buffering, where the smallest fields of view are assigned

greater bandwidth (See FIG. 1). In a simplistic example, a 360 video stream may be split into 4 fields of view, called quadrants. The first quadrant is defined as the smallest and current field of view of the user. The second quadrant is to the user's right, the third quadrant is directly behind the user, and the fourth is to his left. The client, when requesting video, informs the server of its angle of orientation. The server then chunks the next frame of 360-degree video it will stream down into quadrants, with the first quadrant being centered around the last known viewing angle of the user, quadrant #2 being to the user's right, #3 being behind the user, and so on

When buffering, the server, which would normally pick a video quality (for example, low/medium/high) for the next frame based on the amount of available bandwidth the server perceived during the downloading of the previous frame, instead has more degrees of freedom to choose separate video qualities for the likely-to-be-in-view vs. the likely-to-be-out-of-view angles. This optimization assumes "inertia" -- i.e. assumes that a user is most likely to be in the same orientation he is currently in at a future video frame, and is least likely to be facing 180 degrees reversed. For example if a 360 degree video is available in quality of either "low" 400kbps or "high" 800kbps, but recent video frames that were downloaded by the client indicate that only 700 kbps of bandwidth is reliably available, this system can stream the 3 quadrants in front of the user at high quality, and the rear quadrant (highest view angle) in low quality to achieve 700kbps. With each additional frame that is streamed, the frame of reference is re-defined, centered around the user's current viewing angle. Small numbers of quadrants (like 4) need not be used - the 360 degree video can be chunked into a large number of different viewing angles that can be streamed at different qualities. Any type of streaming 360 degree video provider would likely benefit from this type of optimization.



360 Degree Streaming Video

FIG. 1