Ultra Wide View Based Panoramic VR Streaming

Ran Ju, Jun He, Fengxin Sun, Jin Li, Feng Li, Jirong Zhu, Lei Han
Huawei Techologies
Nanjing, China 211100
{juran,hejun32,sunfengxin,mark.lijin,frank.lifeng,jirong.zhu,phoebe.han}@huawei.com

ABSTRACT

Online VR streaming faces great challenges such as the high throughput and real time interaction requirement. In this paper, we propose a novel ultra wide view based method to stream high quality VR on Internet at low bandwidth and little computation cost. First, we only transmit the region where user is looking at instead of full 360° view to save bandwidth. To achieve this goal, we split the source VR into small grid videos in advance. The grid videos are able to reconstruct any view flexibly in user end. Second, according to the fact that users generally interact at low speed, we expand the view that user requested to meet the real time interaction requirement. Besides, a low resolution full view stream is supplied to handle exceptional cases such as high speed view change. We test our solution in an experimental network. The results show remarkable bandwidth saving of over 60% in average at little computation cost while supplying the same quality of experience as local VR.

CCS CONCEPTS

• **Networks** → *Overlay and other logical network structures*; *World Wide Web (network structure)*;

KEYWORDS

Virtual reality, video streaming, low latency

ACM Reference format:
Ran Ju, Jun He, Fengxin Sun, Jin Li, Feng Li, Jirong Zhu, Lei Han. 2017. Ultra Wide View Based Panoramic VR Streaming. In *Proceedings of VR/AR Network '17, Los Angeles, CA, USA, August 25, 2017,* 5 pages.
https://doi.org/10.1145/3097895.3097899

1 INTRODUCTION

The tide of VR has swept over the world recently. By creating a virtual 3D environment which can be interacted with in real time, VR supplies entirely fresh experience to users, thus is thought as the next generation video, game, and social media. Generally, VR content can be produced by computer graphics, multi-view, light field and panoramic photographing. The 360° panoramic video is preferable and most popular because it is cheap and easy to produce, distribute and consume. While VR devices such like panoramic camera and HMD (Head Mounted Display) become increasingly mature, there have been also rich VR resources on Internet.

The popularity of VR, especially the 360° panoramic video, rises naturally the demand of online distribution. However, traditional video streaming are facing great challenges for VR. First, panoramic VR has a much higher throughput because it supplies much larger view angle than traditional displays like LCD TV (e.g. 100° × 100° versus 30° × 20° typically). Current panoramic VR with 4K resolution (3840 × 1920) and over 30Mbps bitrate are quite popular on Internet, which are beyond the capability of access bandwidth in many countries or regions over the world. Worse, the throughput of VR will grow higher rapidly in the future for better experience. The second challenge for Internet VR is the low latency requirement. VR video is displayed in an interactive manner, where the screen changes instantly according to users attitude. To supply a good immersive experience, the motion to photon (MTP) delay, that is, the time from user's movement to that screen refreshed to match the movement, is suggested to be less than 20ms [1] to avoid dizziness and nausea.

Recently a few efforts have been made to overcome the above challenges. A simple and direct solution is to stream the full 360° video to users. Since the full view data is cached in user end, the second challenge, i.e. the 20ms MTP delay, is easy to meet. However, the full view streaming solution has a quite low bandwidth utilization because only a small part of the stream, that is, the Field-of-View (FoV), is consumed by end user. Quantitatively, given an HMD of 100° × 100° FoV, it only takes 21.28% of the full view. A smarter choice is to transmit the FoV region instead of 360° panorama to save bandwidth. However, this solution calls for powerful computation to perform real time FoV encoding, and strict low latency to meet the 20ms MTP requirement. Both the two conditions are difficult to satisfy in current servers and Internet.

In this paper, we propose an ultra wide view based solution to stream high quality panoramic VR on Internet at low bandwidth and computation cost. Our solution is based on two basic observations. First, the angular speed of human's view change is relatively low [3], which implies an expanded FoV streaming can relax the network delay effectively. For example, typically the angular rate of human's head rotation is below $100°/s$, which indicates an

VR/AR Network '17, August 25, 2017, Los Angeles, CA, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5055-6/17/08...$15.00
https://doi.org/10.1145/3097895.3097899

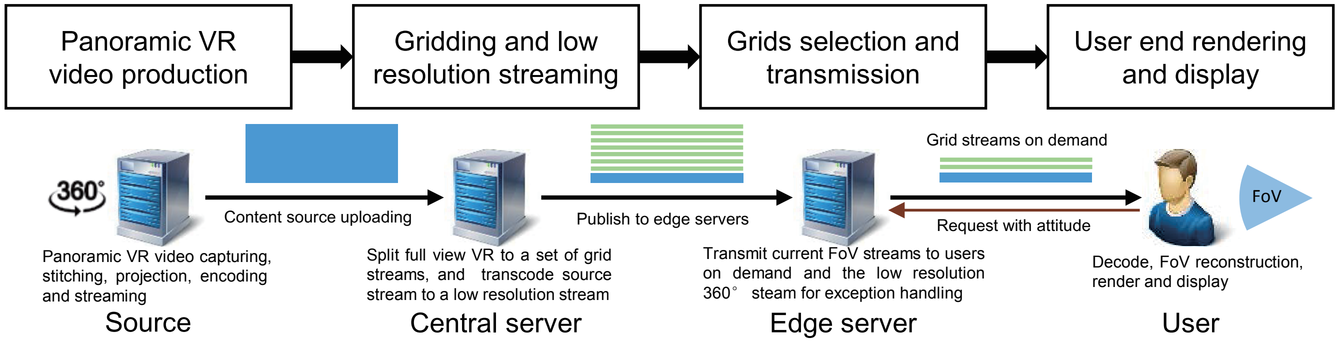
Panoramic VR video production
Gridding and low resolution streaming
Grids selection and transmission
User end rendering and display
360°
Content source uploading
Publish to edge servers
Grid streams on demand
Request with attitude
FoV
Panoramic VR video capturing, stitching, projection, encoding and streaming
Split full view VR to a set of grid streams, and transcode source stream to a low resolution stream
Transmit current FoV streams to users on demand and the low resolution 360° steam for exception handling
Decode, FoV reconstruction, render and display
Source
Central server
Edge server
User

Figure 1: Overview of the proposed solution.

extra 10° view takes 100ms to move out. Second, human eyes perceive blurry images in movement [9]. It is unnecessary to supply high quality display in high speed view change. Owing to this fact, a low quality streaming is supplied to deal with high speed view change and other abrupt cases. According to the observations, our solution employs expanded FoV to achieve efficient VR streaming. The expanded FoV stream is generated by splitting full view into grids in advance and reconstructed in user end. Besides, a low quality full view stream is supplied for each user to handle accidental cases.

We implement our solution in an experimental network with 3 servers and 6 users. 4 panoramic VR videos downloaded from Internet and a live broadcasting platform are used for evaluation. The experimental results show that our solution can effectively reduce the bandwidth by about 60% in average compared with full view streaming, while supplying the same quality of user experience. The extra storage cost is about 1.3 times that of full view streaming solution in average. The results suggested that our solution is capable of supplying efficient VR delivery on Internet.

The rest of this paper is organized as follows. In Sect. 2 we give a detailed introduction of the proposed solution. Then we give the experimental results and some discussions in Sect. 3. After that, we give a review and comparison to previous works in Sect. 4. At last, we give the conclusion and a few remarks of the future work in Sect. 5.

2 METHOD

The overview of our solution is illustrated in Fig 1. Suppose a 360° VR video is captured on the source node, it is first processed to produce the panoramic video by image stitching, sphere projection, video encoding etc. Then the source video is uploaded to the central server. Next, it is split and transcoded into small grid streams. Each grid stream covers a small view angle and the view angles of different grids are approximately equivalent to facilitate reconstruction. For simplicity each grid is encoded independently. Once requested from a user, the edge server analyzes the user's attitude and calculates which grids should be selected to reconstruct the user's FoV. Then a few selected grid streams are transmitted to user end. Besides, a low resolution (LR) full view stream is generated from the source VR video and published to all the edge nodes. Edge servers keep on transmitting the LR 360° stream to users to deal with exceptions. Specifically, in abrupt cases such as Internet congestion, ARQ, high speed view change and so on, the LR 360° stream will feed to the render pipeline to guarantee user experience until new grid streams arrive. For live broadcasting, all the above procedures are performed in real time.

2.1 Panoramic VR Video Gridding

A key step in our solution is to split the source VR video into small grids to enable reconstruction of any FoV image. For feasibility the grid should be as small as possible. However, to improve encoding efficiency the grid should not be too small for intra-frame reference. Owing to these considerations, we make a trade-off between feasibility and coding efficiency. Besides, each grid should have about the same view angle so that any FoV covers nearly constant number of grids. For simplicity, we project the 360° image onto a sphere and mesh the surface based on cube subdivision. The grid streams are denoted as $s_G = \{s_0, s_1, ..., s_{n-1}\}$, where n is the number of grids. Typically, we mesh the sphere with 900 grids where the size of each grid covers about $6^\circ \times 6^\circ$ view angle. Each grid is mapped to a 64×64 pixels square on image plane for a 4K video. According to our experiments, after gridding the total size of grid streams increases by 10% to 50% compared to the source VR video. The gridding is performed in the central server. Besides, the central server transcodes the source VR video s_H to a low resolution (LR) stream s_L, where s_L is one-fourth downsampled from s_H typically. Then, all the grid streams with the LR stream, i.e. $\{s_G, s_L\}$, are delivered to edge servers.

2.2 Grid Streaming on Demand

By source VR gridding, the edge server is able to deliver a part of grid streams instead of all to save bandwidth. Specifically, the user first reports the view angle parameter $\{w, h\}$ of the HMD device to the edge, which indicates the horizontal and vertical angles respectively. Then, as shown in Fig. 2, the user device sends requests carrying user's head

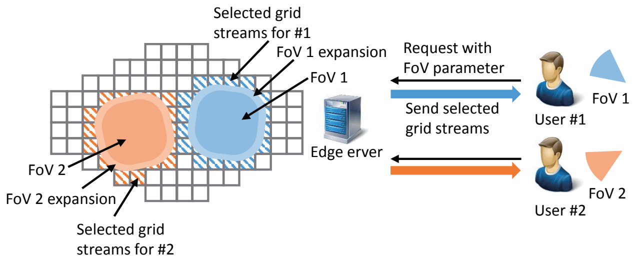
Selected grid streams for #1
FoV 1 expansion
FoV 1
Request with FoV parameter
Send selected grid streams
Edge erver
User #1
FoV 1
User #2
FoV 2
FoV 2
FoV 2 expansion
Selected grid streams for #2

Figure 2: Grid streaming on edge server. Upon the user requests with FoV parameters, edge server expands the FoV region and selects intersected grids. The grid streams are then delivered to user end.

attitude to the edge in real time. The attitude parameter is a triplet $\{a_x, a_y, a_z\}$ that encodes the 3D rotation angle of user's view. According to the parameters, the edge server calculates the set of grids that intersect user's FoV. The intersected grid streams are transmitted to user end.

Exact FoV requires very low network latency to meet the 20ms MTP delay. Considering the time for rendering and display, there are only few milliseconds left for transmission. It is difficult to meet such critical requirements on Internet. To overcome this problem, we employ motion prediction to relax the transmission delay. By predicting user's movement and transmitting the data in advance, the data cached at user end is able to meet the requirement of view change in a short period. As a result, the network delay can be relaxed to several hundreds of milliseconds. A simple prediction method is to expand user's FoV in all directions. Then we get a view angle of $\{w + \theta, h + \theta\}$ where θ is the angle of expansion. Suppose user's rotation angular rate is ω, the time that the user moves out of the expanded FoV region is $\frac{\theta}{\omega}$. Typically ω is below $100°/s$. If we set θ as $10°$, it relaxes the network delay to nearly 100ms at the cost of 17.6% extra view.

In the user end, the grid streams are synchronized and decoded. Then the grid frames are rendered in the same space to reconstruct user's FoV image. When view change happens, user end will first load local expanded FoV data for display. Meanwhile, the user sends a request with new attitude parameter to edge server. Then the edge server will select a new set of grid streams and transmit them to user end. The new set of streams are expected to arrive before the local data exhausted or user's view moved out of the expanded FoV region. So the expansion angle θ and local caching length L can be calculated according to:

$$\alpha\frac{N}{R} + T_P + T_S \leq \min(\frac{\theta}{\omega}, L) \qquad (1)$$

where N is the size of a minimal block which can be decoded independently, e.g. GOP (Group of Pictures). R is the transmission rate. T_P and T_S are the propagation delay and processing time respectively. α is a jitter factor estimated from historical record.

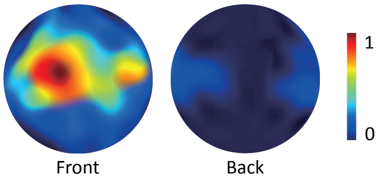

Figure 3: Heat map of users' attention. Regions appear hotter indicate that users are more likely to look at.

2.3 Low Resolution Full View Backup

In some exceptional situations, e.g. link congestion, key frame packet loss, high speed view change and so on, we supply a full view low resolution stream to guarantee user experience. The low resolution stream is typically downsampled by $\frac{1}{4}$ from the source VR stream, and transmitted to user end together with the grid streams. When user view moves too fast and out of the expanded FoV, the user end will load the low resolution stream to display. Meanwhile, the user requests for new view and the server starts delivering the requested data. When the new FoV data arrives and synchronizes with user end, the user will switch to grid streams as input for VR display. The low resolution full view stream could also improve the efficiency for grid streams encoding. Since the grid streams are encoded independently without intra-frame reference, the low resolution stream supplies a reference for grid streams encoding.

2.4 Acceleration for Live Broadcasting

We observed that users tend to look at few hot spots in VR watching. A statistical heat map is illustrated in Fig. 3, where hot color indicates high probability of attention. According to the heat map, we may conclude that users are more likely to look at low latitudes and favorite the front. This can be explained by that people tend to watch with a comfortable gesture in VR watching, i.e. looking at the front or scanning horizontally. According to this observation, a multi-path delivery strategy is utilized for live broadcasting acceleration. We discriminatively deliver grid streams that have high attention probability in a faster way. Specifically, hot grid streams are assigned with high priorities and transmitted via fast path by our private transport protocols. Besides, redundant delivery [6, 14] is an optional choice for hot streams acceleration.

2.5 Implementation Details

In real applications there are still some engineering problems to be solved. We further give a few implementation details as follows.

(1) At the request initiated from an end user, the server starts to send the FoV stream and the LR full view stream. The LR stream is generated from the original panoramic video. The resolution of the LR stream is smaller than the FoV stream and typically set as one

quarter of the source resolution. The user attitude is carried by the request. Two buffers are used to receive the two streams. The end device starts to play when the buffer is ready.

(2) The headset decodes and plays the FoV stream when the user's current attitude is inside the viewport, otherwise the LR stream is used for display. Note during the user's movement, the FoV and the LR stream could be stitched for display. When the user's view totally out of the viewport of the FoV stream, the headset extracts the new view from the LR stream.

(3) The switching from LR to FoV stream is controlled by time stamp synchronization. Once the new FoV data arrives and gets synchronized with the user's play time, the headset switches to the FoV buffer. Otherwise, the headset acquires the LR stream or the stitching of the LR and FoV stream for display. Motion prediction could be used to improve the synchronization. The data out-of-date in the buffer is dropped directly.

3 EVALUATION AND ANALYSIS

3.1 Experimental Settings

We construct an experimental network with 1 central server, 2 edge servers and 6 user terminals, which form a tree logical topology. Each server has a 4GHz Intel i7 6700K CPU, 16GB memory and a GTX 980 GPU. Huawei P9 Plus and Huawei VR headset are selected for end VR display. The bandwidth of the fixed network is set as 100Mbps and the end phone is connected via WiFi. We evaluate 4 panoramic VR videos of 4K resolution downloaded from Internet which cover different cases such as indoor and wild scenes. In live broadcasting we use an insta360 camera to capture panoramic VR, which streams 4K VR at 12Mbps bitrate and 25fps.

3.2 Results and Analysis

In Table. 1 we give the storage and bandwidth consumption compared with full view solution in percentage. A few remarks can be concluded as follows. First, the average bandwidth consumption is reduced by over 60% compared to full view solution, which shows the effectiveness of the proposed method. Second, there is a gap between the minimum and maximum bandwidth consumption. This indicates that visual information is non-uniformly distributed in 360° space. The average bandwidth consumption is higher than the proportion of FoV (21.28%), which implies that the regions users tend to look at encode more visual information. We also show a traffic record of the full view and the proposed solution in Fig. 4. At last, the storage consumption of grid streams is about 1.3 times of the original VR in average. This is explained by that grid streaming weakens the intra-frame encoding efficiency. We will consider to adopt advanced encoding techniques such as low resolution reference to deal with this problem in the future.

Table 1: Quantitative results. The bandwidth and storage consumption compared to full view solution.

	Vod 1	Vod 2	Vod 3	Vod 4	Live
min BW (%)	12.45	11.68	14.50	11.85	10.56
max BW (%)	50.61	42.79	53.84	39.83	46.50
avr BW (%)	30.69	33.67	42.06	31.62	35.30
Storage (%)	145.24	115.56	146.17	126.78	121.51

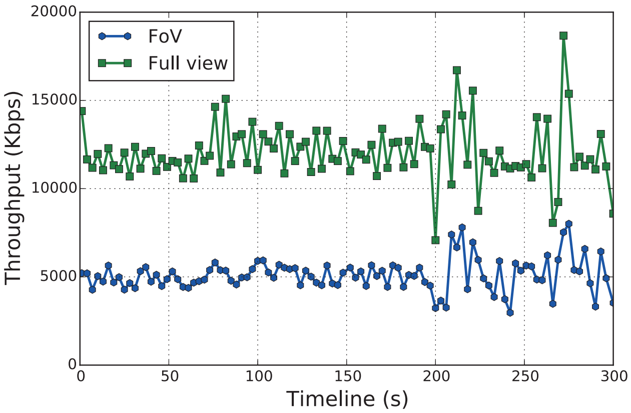

Figure 4: Traffic record of the full view and the proposed solution.

4 RELATED WORK

Full view delivery is a simple solution for VR streaming. Obviously, the shortage is the low utilization of bandwidth since only about 21.28% view is consumed by user end. Recently a few FoV (Field of View) based methods have been proposed [4, 5, 7, 10, 11]. The main idea is to use pyramid, equirectangular or dodecahedron projection [8, 13] to encode a few discrete viewports in advance, and fetch the stream according to user's orientation. These approaches are shown to be effective in bandwidth saving. However, they consume much (about 6 times) storage owing to the high redundancy among different viewports. Besides, the quality in view change is low. Another approach is to employ the power of edge computing [2, 15] to render user's view in real time. However, it consumes very high computation and requires low network latency [12]. In contrast, our approach achieves the same user experience with local VR at low bandwidth, storage and computation cost.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have proposed an ultra wide view based method for online VR streaming. By delivering partial necessary data instead of all, we significantly reduced the bandwidth consumption. To meet the real time interaction requirement, we made use of FoV expansion to relax transmission delay. Video meshing was employed to facilitate feasible reconstruction of any viewports. Besides, a low resolution full view stream was supplied to handle abrupt cases such as high speed interaction. The evaluation on an experimental network showed that our solution is competent for panoramic VR streaming on Internet.

The following points are worthy to study in the future. First, the visual heat map could be well utilized to improve network transmission since most of the 360° view is cold. Second, video quality evaluation could be utilized to score different VR streaming solutions and consequently make user experience improved. At last, we also expect to propose or employ novel encoding methods tailored for panoramic videos. According to our experiment, a well designed VR video encoding which could be split, transmitted and reconstructed feasibly while having good compression rate will be of great help to obtain considerable bandwidth saving.

REFERENCES

[1] Michael Abrash. 2014. What VR could, should, and almost certainly will be within two years. *Steam Dev Days, Seattle* (2014).

[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 13–16.

[3] William Bussone. 2005. *Linear and angular head accelerations in daily life*. Ph.D. Dissertation. Virginia Tech.

[4] Shenchang Eric Chen. 1995. Quicktime VR: An image-based approach to virtual environment navigation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 29–38.

[5] Xavier Corbillon, Alisa Devlic, Gwendal Simon, and Jacob Chakareski. 2016. Viewport-Adaptive Navigable 360-Degree Video Delivery. *arXiv preprint arXiv:1609.08042* (2016).

[6] Jeffrey Dean. 2012. Achieving Rapid Response Times in Large Online Services. (2012). http://research.google.com/people/jeff/latency.html

[7] David Pio Evgeny Kuzyakov. 2016. Next-generation video encoding techniques for 360 video and VR. (2016). https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/.

[8] Chi-Wing Fu, Liang Wan, Tien-Tsin Wong, and Chi-Sing Leung. 2009. The rhombic dodecahedron map: An efficient scheme for encoding panoramic video. *IEEE Transactions on Multimedia* 11, 4 (2009), 634–644.

[9] Li Li, Bernard D Adelstein, and Stephen R Ellis. 2009. Perception of image motion during head movement. *ACM Transactions on Applied Perception (TAP)* 6, 1 (2009), 5.

[10] King-To Ng, Shing-Chow Chan, and Heung-Yeung Shum. 2005. Data compression and transmission aspects of panoramic videos. *IEEE Transactions on Circuits and Systems for Video Technology* 15, 1 (2005), 82–95.

[11] Yago Sánchez, Robert Skupin, and Thomas Schierl. 2015. Compressed domain video processing for tile based panoramic streaming using HEVC. In *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2244–2248.

[12] Shu Shi, Cheng-Hsin Hsu, Klara Nahrstedt, and Roy Campbell. 2011. Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming. In *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 103–112.

[13] Kashyap Kammachi Sreedhar, Alireza Aminlou, Miska M Hannuksela, and Moncef Gabbouj. 2016. Viewport-Adaptive Encoding and Streaming of 360-Degree Video for Virtual Reality Applications. In *Multimedia (ISM), 2016 IEEE International Symposium on*. IEEE, 583–586.

[14] Zhe Wu and Harsha V Madhyastha. 2013. Understanding the latency benefits of multi-cloud webservice deployments. *ACM SIGCOMM Computer Communication Review* 43, 2 (2013), 13–20.

[15] Shanhe Yi, Cheng Li, and Qun Li. 2015. A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*. ACM, 37–42.