Fixation Prediction for 360° Video Streaming to Head-Mounted Displays

Ching-Ling Fan[1], Jean Lee[1], Wen-Chih Lo[1], Chun-Ying Huang[2], Kuan-Ta Chen[3], and Cheng-Hsin Hsu[1]
[1]Department of Computer Science, National Tsing Hua University
[2]Department of Computer Science, National Chiao TUng University
[3]Institute of Information Science, Academia Sinica

ABSTRACT

We study the problem of predicting the Field-of-Views (FoVs) of viewers watching 360° videos using commodity Head-Mounted Displays (HMDs). Existing solutions either use the viewer's current orientation to approximate the viewed tiles in the future, or extrapolate the FoVs using the historical orientations and dead-reckoning algorithms. In this paper, we develop *fixation prediction networks* that concurrently leverage sensor- and content-related features to predict the viewer fixation in the future, which is quite different from the solutions in the literature. The sensor-related features include HMD orientations, while the content-related features include image saliency maps and motion maps. We build a 360° video streaming testbed to HMDs, and recruit twenty-five viewers to watch ten 360° videos. We then train and validate two design alternatives of our proposed networks, which allows us to identify the better-performing design with the optimal parameter settings. Trace-driven simulation results show the merits of our proposed fixation prediction networks compared to the existing solutions, including: (i) lower consumed bandwidth, (ii) shorter initial buffering time, and (iii) short running time.

1 INTRODUCTION

More and more people share their life and experience with friends, relatives, and general publics over online social networks anywhere, anytime. Among the media types exchanged in online social networks, video content is the most appealing and widely-used media. Recently, 360° videos are getting popular, because they preserve *immersive experience*, allowing people to better share their life and experience. A market report predicts that the global market of 360° cameras will grow at an annual rate of 35% between 2016 and 2020 [2]. Such an increase is further boosted by the growing attentions of consumer-grade Head Mounted Displays (HMDs), which provide wide Field-of-Views (FoVs), and come with integrated sensors for determining view orientation and head position. In fact, another market research indicates that the global market of Virtual Reality (VR) related products will reach 30 billion USD by 2020 [1]. Due to their popularity, the commodity HMDs, such as HTC Vive, Samsung Gear VR, and Oculus Rift, are ideal for playing out 360° videos to viewers for even more immersive experience.

(a) (b)

Figure 1: Using HMDs to watch 360° videos allows viewers to naturally navigate through different parts of the videos. However, it comes with some challenges: (a) each viewer only sees a small part of the video at any moment, and (b) existing video codecs dictate rectangle video inputs, and thus require mapping models, leading to distorted images.

Leveraging commodity HMDs for 360° video streaming is, however, very challenging for two reasons illustrated in Fig. 1. Fig. 1(a) reveals that, while the 360° video is in extremely high resolution, with HMDs, each viewer only gets to see a small part of the whole video. Therefore, sending the whole 360° video in full resolution may lead to *waste* of resources, such as network bandwidth, processing power, and storage space. Another way to stream 360° videos to HMDs is to only stream the *current* FoV of the viewer. We emphasize *current*, because the FoV changes as the viewer's head and eyes move, which leads to the following main challenge: *which FoV should we transfer to meet the viewer's needs in the next moment?* This challenge makes designing a *real-time* 360° video streaming system to HMDs quite tricky, especially because most video streaming solutions nowadays deliver videos in segments lasting for *a few seconds* [27].

Most existing studies [14, 19, 23] partially cope with this challenge by a straightforward solution: they stream the current FoV (could be enlarged a bit by a heuristic factor) in the full resolution, and also stream the whole 360° video in a reduced resolution. When a viewer moves his/her head outside of the current FoV (used by the streaming system), the reduced-resolution video is played to the viewer, before the full-resolution one comes (in a few seconds). This approach results in two drawbacks: (i) jumping between full- and reduced-resolutions greatly degrades the user experience [12]; and (ii) sending parts of videos that are far away from the FoV (say at the opposite orientation of the HMD), although in reduced resolution, still wastes precious resources.

NOSSDAV'17, Taipei, Taiwan

DOI: http://dx.doi.org/10.1145/3083165.3083180

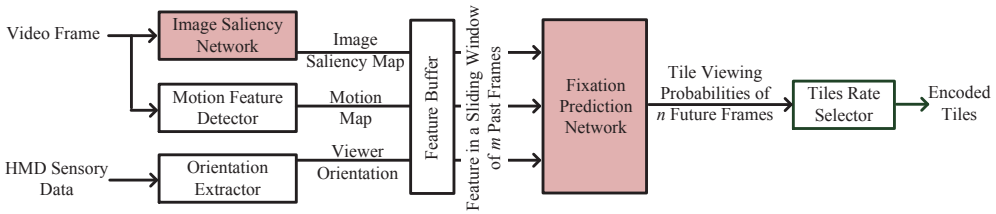


Figure 2: Overview of the proposed 360° video streaming server. A tile streaming example is shown.

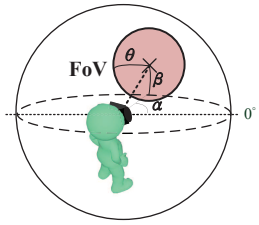


Figure 3: The FoV model.

A better solution is to predict the viewer's *fixation*, which is the center of a viewer's FoV, in the near future, and only transfer the parts of 360° videos that are viewed with high probability. Although recent studies [22, 23] attempt to predict the fixation, they adopt simple *extrapolations*, which are less accurate, and thus result in degraded user experience and wasted resources. *In this paper, we solve the fixation prediction problem using neural networks for higher accuracy.* The unique feature of our fixation prediction networks is that we concurrently consider *content-* and *sensor-related* features. Content-related features include image saliency map [5] and motion map [18], and sensor-related features come from the orientation and position sensors on HMDs. To the best of our knowledge, neural networks that incorporate both content- and sensor-related features, have not been studied in the literature.

Predicting fixation is no easy task. Although Convolutional Neural Network (CNN) has been widely used in video (and image) domain [13], most prior studies focus on videos shot by conventional cameras and may not work well with 360° videos. This is because the 360° videos are *mapped* before being encoded, as the existing video codecs only support *rectangle* videos. Several mapping models, such as cube and pyramid, are explored in both the industry [3] and the academia [10]. These mapping models, unfortunately, cause some distortion on the shape of objects. Take the well-known *equirectangular* model as an example, Fig. 1(b) shows that the objects close to the north and south poles are seriously distorted, which may negatively affect the effectiveness of the pre-trained CNNs [9]. In this paper, we report the performance of our proposed fixation prediction networks after considering such distortion.

We build a 360° video streaming testbed. We recruit twenty-five viewers to watch ten 360° videos downloaded from YouTube. We use parts of the collected dataset to train and validate our proposed fixation prediction networks for optimal parameter settings. We then conduct trace-driven simulations to evaluate the performance of our solution. Compared to the existing solutions in the literature, our solution: (i) delivers comparable video quality, (ii) requires shorter initial buffering time (up to 43% reduction), (iii) consumed less bandwidth (a 22-36% reduction), and (iv) completes in real time (≤ 50 ms). The evaluations results show the potential of our fixation prediction networks on enhancing the 360° video streaming to HMDs.

2 RELATED WORK

Viewer fixation prediction. For prediction with content-related features, salient object detection has been studied [5] for still images. Mavlankar and Girod [23] perform fixation prediction in videos using features like thumbnails, motion vectors, and navigation trajectory expolation. With the advanced machine learning technologies, various supervised learning methods including neural networks are adopted for better feature extraction and prediction accuracy in fixation detection [4, 7, 24]. Chaabount et al. [7] build a CNN architecture and use residual motion as the features for predicting saliency in videos. Alshawi et al. [4] observe the correlation between the eye-fixation maps and the spatial/temporal neighbors, which provides another way to quantify viewer attention on videos. Nguyen et al. [24] propose to adopt the information of static saliency (in images) and then take camera motions into considerations for dynamic saliency (in videos) prediction. These studies [4, 5, 7, 23, 24] do not focus on 360° video streaming to HMDs, and fail to take sensor-related features into account.

Commodity HMDs. Several commodity HMDs are released and their applications has gotten increasingly popular. Their performance has been recently studied in the literature. Young et al. [30] perform subjective tests using commodity HMDs and find that they work well for playing out videos (one-way) and supporting interactive tasks (two-way). Friston and Steed [11] present a frame counting mechanism for synchronization, in order to measure the latency of HMDs. Chang et al. [8] propose methodologies to measure the latency, sensitivity, and precision of HMDs. Several tradeoffs are observed on commodity HMDs in their study. These studies [8, 11, 30] do not try to identify the viewer's FoV, let alone predicting viewer fixations.

3 OVERVIEW

3.1 360° Streaming Systems

Fig. 2 presents our proposed architecture of a 360° streaming server, in which we focus on the software components related to *fixation prediction*. We have identified two types of content-related features: image saliency map [5] and motion map [18]; and sensor-related features, such as orientations and angular speed, from HMDs. We briefly describe these components in the following:

- **Image saliency network** is a deep neural network trained to predict the *image saliency map*, which shows the parts of the image that attract viewers the most.
- **Motion feature detector** analyzes the Lucas-Kanade optical flow [20] of consecutive frames, because viewers may be attracted by moving objects.

- **Orientation extractor** derives the viewer orientation data[1], including yaw, pitch, and roll, from HMD sensors.
- **Feature buffer** stores the features, including saliency map, motion map, and view orientation in a sliding window, which are used for fixation prediction.
- **Fixation prediction network** uses content-related (image saliency map and motion map) and sensor-related (viewer orientation) features as inputs to predict future viewing probability of each tile[2].
- **Tile rate selector** performs rate allocation among video tiles.

The interactions among these components are as follows. The video frames are sent to the image saliency network and motion feature detector for generating the image saliency map and the motion map, respectively. Generating these two maps is potentially resource demanding, and we assume that they are created offline for pre-recorded videos. The HMD sensory data are transmitted to the orientation extractor to derive the viewer orientations. The feature buffer maintains a sliding window that stores the latest image saliency maps[3], motion maps, and viewer orientations as the inputs of fixation prediction network. The video fixation prediction network predicts the future viewing probability of each tile. The tile rate selector optimally selects the rates of the encoded video tiles [4] The shadowed components in Figure 2 are the focus of this paper and will be detailed in Sec. 4.

3.2 Field-of-View and Mapping Models

We conduct experiments of playing a 360° video with artificial *grids* to viewers, and collect questionnaires to understand how to model the FoV of commodity HMDs. We find that the FoV of existing HMDs, including Oculus Rift, HTC Vive, and Samsung Gear can all be modeled as a circle on sphere[5]. Fig. 3 presents the FoV model of HMD. The viewer stands at the center of the sphere. Let α and β be the yaw and pitch, which are reported from the sensors equipped by HMDs. Knowing α and β, we can find the center point that the viewer looks at. Furthermore, we let θ be the radius of FoV in degree. Therefore, we describe the FoV in the spherical space as $f_s = (\alpha, \beta, \theta)$. The measured θ values are about 100° (Oculus Rift), 67° (HTC Vive), and 67° (Samsung Gear). We use 100° in our experiments if not otherwise specified. We note that this is a basic model, which does not capture the HMD positions. This is not a concern, because most 360° video streaming systems ignore viewer positions. Moreover, recent HMDs, such as FOVE, start to support eye-tracking, which may lead to new type of sensory data. Our approach is readily extensible to such HMDs once they hit the market.

There are several mapping models for 360° videos [10, 29], including: (i) equirectangular, (ii) cube, and (iii) rhombic dodecahedron mapping. In this work, we adopt the equirectangular mapping model, which is the most popular and widely supported mapping model for 360° videos. For example, YouTube only supports 360° videos encoded with the equirectangular mapping model. The equirectangular mapping projects the sphere to a *cylinder*. It introduces large distortion at the areas close to poles (see Fig. 1(b)), which may result in redundant data transmission and inferior image saliency detection accuracy.

4 FIXATION PREDICTION NETWORKS

Image saliency network. Different methods to generate the image saliency maps could be classified into two groups. One group (manually) selects low- and high-level image features, such as color, intensity, orientation (low-level) and semantic information (high-level) [17]. The other group constructs deep CNNs to generate image saliency maps [28]. To be *data-driven*, we adopt *pre-trained* CNNs [25] for learning the image features, which were *originally* used for object detections and image classifications. In particular, Cornia et al. [9] propose a deep multi-level network combining weight features from different levels of the CNN, which achieves superior performance by considering low- to high-level features. We adopt this deep multi-level network, whose architecture is based on VGG-16 [25]. Three layers are extracted from the VGG-16 model and combined with two additional layers to improve the generalization for the final image saliency maps. For each video frame, we obtain its image saliency map from the output of this deep multi-level network as a feature.

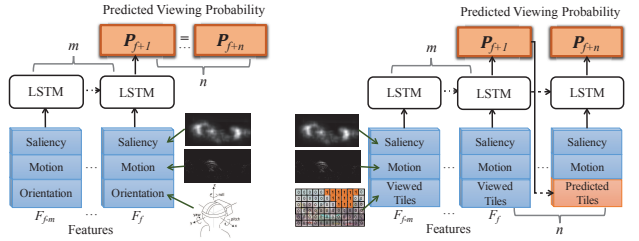


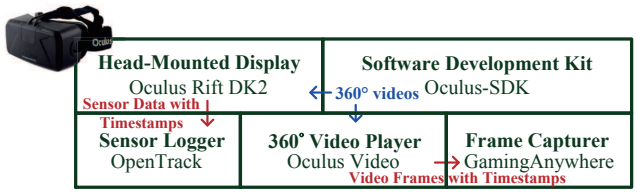
Figure 4: Our proposed fixation prediction networks: (a) orientation-based network and (b) tile-based network.

Fixation prediction network. The function of this network is to predict where the viewer would most likely to look at in the future video frames given a sequence of observed video frames. The fixation prediction network is based on a Recurrent Neural Network (RNN), which is suitable to learn useful information from a time series of video frames. We chose the LSTM (Long Short Term Memory) network [15] to learn more long-term dependencies among video frames. Fig. 4 presents our two proposed fixation prediction networks. Both networks take features of m past video frames in a *sliding window* as inputs, and predict the viewing probability of tiles of n future video frames in a *prediction window* as outputs. Let F_f be the features of frame f. The features include the image saliency map, motion map, viewer orientation, and viewed

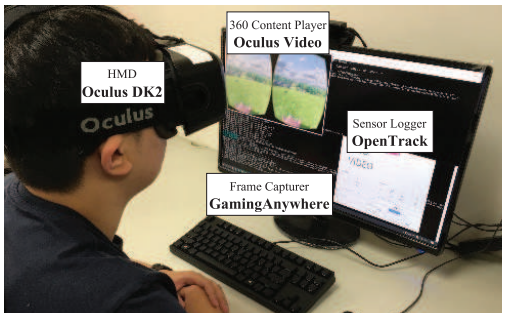
[1] We use the word *orientation* loosely to indicate features that can inferred from HMD sensors. Our current implementation only considers, but can be readily extended to other features.
[2] For the sake of discussion, we consider tile streaming systems, while our fixation prediction networks work for real-time transcoding-based systems as well.
[3] Note that the saliency map we currently considered are the whole frame in equirectangular projection. We believe that considering the whole frame provides more global information than only considering the current view of the viewer.
[4] In general, the tile rate selector could select the encoded rates of each tile. While in Sec. 6, we fix the Quantization Parameter (QP) and resolution of each tile to 28 and 192x192, respectively, for brevity. A larger-scale simulation will be conducted in the future.
[5] The FoV could be different shapes on 2D planes depending on mapping models. For example, it would be ellipses on equirectangular plane.

tiles. Let $p_f^t in [0, 1]$ be the predicted viewing probability of tile t of frame f. We collectively write probability of all tiles of frame f as $\mathbf{P}_f$. The orientation-based network (Fig. 4(a)) takes the view orientation as inputs, and predicts the viewing probability of the next frame. It then *reuse* the same prediction for all the n frames in the prediction windows. In contrast, the tile-based network (Fig. 4(b)) takes the viewed tiles as inputs, and predicts the viewing probability of the next n frames. Note that, the viewed tiles in the future frames are *not* ground truth; instead the predicted viewed tiles are used as indicated by the top-down vertical arrows.

5 DATASET COLLECTION AND NETWORK TRAINING



(a)



(b)

Figure 5: Our testbed for dataset collections: (a) system architecture and (b) a photo of a subject performing experiments.

5.1 Testbed and Datasets

At the time of writing, there exist no public HMD sensory dataset; the ones used in the literature [21, 29] are proprietary. Therefore, we set up a complete 360° video streaming testbed and collect our own dataset as follows. Fig. 5(a) presents the design of our testbed, which consists of: (i) an Oculus Rift, (ii) Oculus Software Development Kit (SDK), (iii) 360° video player (rendering 360° in both HMD and a mirrored screen), (iv) sensor logger based on OpenTrack, and (v) frame capturer based on GamingAnywhere [16].

When a viewer watches a 360° video as shown in Fig. 5(b), the rendered video is captured by frame capturer and stored to disk. The viewer's head movements, including position and orientation, are also recorded by sensor logger. Both of them are timestamped on the same computer. By aligning sensory data and 360° videos, we identify where the viewer watches at any moment.

We download 10 360° videos from YouTube, which are in 4K resolution with a frame rate of 30 Hz. The videos can be classified into three groups: (i) Computer-Generated (CG), fast-paced, (ii) Natural Image (NI), fast-paced, and (iii) NI, slow-paced. Each video lasts for one minute. We recruit 25 viewers for dataset collections. Most of them are in their early twenties, and 64% of them are male. 96% of the subjects seldom use HMDs, while 60% of them use HMDs for the first time. We play all 10 videos to each viewer, which result in 250 *traces* in our dataset. By trace, we refer to a combination of a viewer and a video, in the rest of this paper. Since most subjects are not regular HMD users, collecting the dataset is time consuming: each subject took us 30 mins on average.

5.2 Network Training and Validation

Table 1: Performance of the Orientation-based Network

Model Parameters			*Training Set*			*Validation Set*		
No. Neu.	*LSTM Layers*	*Drop.*	*Rank. Loss*	*Accuracy*	*F-Score*	*Rank. Loss*	*Accuracy*	*F-Score*
256	*2*	*T*	0.12	87.03%	0.65	0.17	84.69%	0.58
256	*2*	*F*	0.11	87.68%	0.66	0.17	84.87%	0.59
256	*1*	*F*	0.11	87.97%	0.66	0.16	85.42%	0.59
256	*1*	*T*	0.1	88.20%	0.67	0.15	85.72%	0.60
512	**1**	**T**	**0.09**	**89.25%**	**0.70**	**0.14**	**86.35%**	**0.62**
1024	*1*	*T*	0.09	89.28%	0.71	0.14	86.06%	0.62

Table 2: Performance of the Tile-based Network

Model Parameters			*Training Set*			*Validation Set*		
No. Neu.	*LSTM Layers*	*Drop.*	*Rank. Loss*	*Accuracy*	*F-Score*	*Rank. Loss*	*Accuracy*	*F-Score*
256	*1*	*F*	0.14	86.09%	0.54	0.20	83.76%	0.50
256	*1*	*T*	0.15	86.12%	0.54	0.20	83.84%	0.50
256	*2*	*T*	0.14	86.37%	0.55	0.20	83.88%	0.51
256	*2*	*F*	0.14	86.58%	0.57	0.20	83.94%	0.52
512	*2*	*F*	0.13	86.91%	0.58	0.19	84.11%	0.52
1024	**2**	**F**	**0.12**	**87.29%**	**0.60**	**0.19**	**84.22%**	**0.53**

We consider the fixation prediction on tiles as the multi-label classification problem and have implemented the neural network using Scikit-Learn and Keras. The ground truth of the fixation prediction networks are the tiles viewed by the viewers at each frame. Using the dataset collected above, we sample the points within the FoV by mapping the orientation on the sphere to equirectangular model. Then, the viewed tiles are the tiles that are overlapped with the mapped points. For a single video frame, each tile is either watched or not, i.e., it has a boolean viewing probability.

We did not retrain the image saliency network. We use the traces from 12 viewers to train the two proposed fixation prediction networks. Among the traces, we randomly divide them into 80% and 20% for training and validation, respectively. The networks are trained to minimize the *logarithmic loss*, also known as *cross-entropy loss*, using Stochastic Gradient Descent [6] with a learning rate of 10^{-2}. We let both the sliding window size m and prediction window size n to be 30. To obtain the optimal parameters, we consider the number of neurons in {256, 512, 1024}, the number of LSTM layers in {1, 2}, and the dropout in {true, false}. We first fix the number of neurons at 256 and train the networks with all combinations of the number of LSTM layers and dropout. After the optimal combination of the number of LSTM layers and dropout is determined, we try different numbers of neurons to find its optimal settings.

We note that the predicted probability is a real number between 0 and 1, and we use a *threshold* ρ to round it to a boolean decision. We refer to $p_f^t \geq \rho$ as *predicted tiles*, and the actually viewed tiles as *viewed tiles*. We let $\rho = 0.5$ if not otherwise specified. To select the optimal parameters of the two models, we consider three metrics: (i) *accuracy*, which is the ratio of correctly classified tiles to the union of predicted and viewed tiles, (ii) *F-score*, which is the harmonic mean of the precision and recall, where the precision and recall are the ratios of correctly predicted tiles to the predicted and viewed tiles, respectively, and (iii) *ranking loss*, which is the number of tile pairs that are incorrectly ordered by probability normalized to the number of tiles.

We give the training and validation results of the two models in Tables 1 and 2, respectively. We find that the optimal model parameter from them are <512, 1, True> and <1024, 2, False>, respectively. We notice that both models perform reasonably well, while the orientation-based network performs slightly better (e.g., $\sim$ 2% higher in accuracy). Hence, we adopt the orientation-based network as our fixation prediction network in the rest of the paper.

6 POTENTIAL OF OUR FIXATION PREDICTION NETWORKS

6.1 Simulation Setup

Table 3: Average Statistics of 4-sec H.265 Video Titles

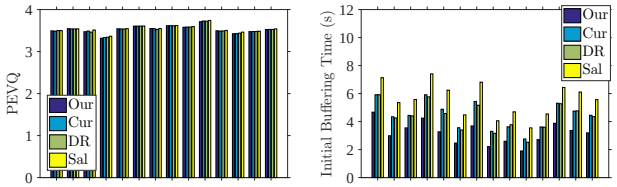
Videos	*Size (B)*	*PSNR (dB)*	*SSIM*	*PEVQ*
CG, Fast-Pace	30662	57.34	0.92	3.55
NI, Slow-Paced	45062	45.02	0.92	3.35
NI, Fast-Pace	21015	47.28	0.96	3.83

We have implemented a simulator in Python to quantify the potential of our fixation prediction networks, where a server supports V concurrent viewers. Each viewer randomly selects a 360° video to watch, and all viewers share a bottleneck link of bandwidth B Mbps. We assume the streaming system has a latency of D secs, and divides videos into s-sec segments with a tile size of 192×192. Each viewer connects to the server and requests for the tiles predicted by m past frames, while n is determined by s and D. Each simulation lasts for 1 min. We consider the first m frames as *warm-up* frames, i.e., their statistics are not accounted for.

Our fixation prediction networks are unique, because we jointly consider content- and sensor-related features. For comparisons, we also implement three other solutions: (i) *Current* (Cur), which uses the current orientation to get the predicted tiles for the next segment, (ii) *Dead Reckoning* (DR) [23], which uses the velocity of viewer orientations to predict the tiles, and (iii) *Saliency* (Sal), which chooses tiles with average saliency values higher than a percentile of λ. Other than accuracy and F-score, we also consider the following performance metrics: (i) *missing ratio*, which is the fraction of missing tiles over all viewed tiles, (ii) *consumed bandwidth* of each viewer, (iii) *initial buffering time*, which is the minimum buffering time for smooth playout, (iv) video quality in Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Perceptual Evaluation of Video Quality (PEVQ) [26], and (v) running time of predicting tiles. For all 360° videos, we cut each frame into 192×192 tiles, where each tile lasts for 4 secs. We then encode each tile using x265 with the default settings and a Quantization Parameter (QP) of 28. Table 3 reports the average quality levels and tile size across all tiles and videos in three metrics. This table shows the encoded tiles have reasonable quality levels.

Some pilot simulation runs with $V = 13$ viewers[6], $B = 150$ Mbps, $D = 2$ secs, and $m = 30$, reveal that the missing ratios are nontrivial for our and baseline solutions, when $\lambda = 99\%$ and $\rho = 0.5$: 30+% missing ratios are observed. To be *practical*, we augment our and baseline solution to ensure a sub-10% average missing ratio by adjusting ρ and λ. For Current and DR, we iteratively add new tiles at the edge of predicted tiles for δ times to accommodate the inferior missing ratio. The resulting ρ is 0.05, λ is 5%, and δ's for Current and DR are 4.

6.2 Results



(a) (b)

Figure 6: Compared to the existing solutions, our fixation prediction networks: (a) result in comparable video quality and (b) requires shorter initial buffering time.

Service quality of our fixation prediction networks. The average missing ratios of our solution and the three baseline solutions (Cur, DR, and Sal) are 9.2%, 5.8%, 8.1%, and 0.2%, respectively. These ratios meet the 10% target, and are comparable. Moreover, we plot the perceived video quality of all 13 viewers as well as the average values in Fig. 6(a), which shows that the video quality in PEVQ of received tiles are also comparable. We note that the same observation can be made when other video quality metrics (PSNR and SSIM) are used; their figures are not shown due to the space limitations. Fig. 6(b) reports the required initial buffering time to guarantee that all tiles will arrive in time. This figure clearly shows that our solution needs shorter initial buffering time. This can be attributed to the flexibility provided by our proposed solution. Instead of only enlarging the range around the predicted tiles (Cur and DR) and lacking of the information of the current view (Sal), our proposed solution transmits the tiles that have higher viewing probabilities More concretely, while other solutions need as high as 5.57 secs, our solution only needs 3.20 secs; a reduction of 43%.

Overhead of our fixation prediction networks. Fig. 7(a) plots the bandwidth consumed by individual users and the average bandwidth consumptions of our and other solutions. This figure demonstrates that our solution does *not* trade high resource (bandwidth) consumption for good service quality. Instead, our solution consumes less bandwidth compared to the baseline solutions. More precisely, on average, each user only consumes 7 Mbps in our solution; each user consumes 9 Mbps with Cur and DR, and

[6]We use all the testing viewers in our dataset.

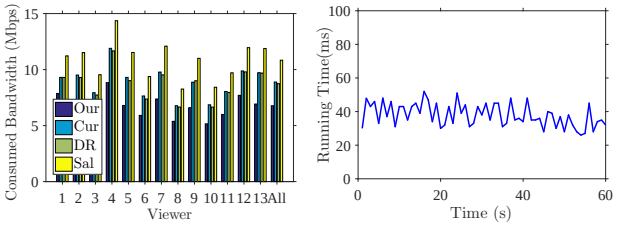


Figure 7: Our fixation prediction networks: (a) consume less bandwidth than the existing solution and (b) runs in real-time.

11 Mbps with Sal; that is our solution reduces the bandwidth consumption by 22-36% on average. Fig. 7(b) gives the running time of generating the tile predictions. This figure clearly show that the tile prediction can be done in 50 ms, which is negligible compared to the segments that last for several (say 4) seconds.

7 CONCLUSION

In this paper, we address the problem of fixation prediction for 360° video streaming to HMDs using two neural networks. The novelty of our solution is that we concurrently leverage content- and sensor-related features, which has never been proposed to our best knowledge. To develop our fixation prediction networks, we build a real testbed and recruit 25 viewers to watch 10 real 360° videos. We clean up the logged sensory data and analyzed video data into a dataset, and use it to train our proposed fixation prediction networks for optimal parameter settings. Through trace-driven simulations, we find that compared to the existing solution, our solution consumes less bandwidth, requires shorter initial buffer time, and runs in real-time.

We acknowledge that the current work can be extended in several dimensions. For example, more extensive simulations are needed to better understand the merits and limitations of our fixation prediction networks. Those simulation results may help us to enhance the design of our solution, but couldn't be included in the current paper due to the space limitations. A larger-scale dataset could provide more reliable results. Furthermore, training and testing data can be improved by splitting them depending on the video types. This could further prove our network works well given new video content. Another open issue is to understand how other 360° video mapping models, such as the cube and rhombic dodecahedron [10, 29], (negatively) affect the performance of the pre-trained CNNs (which were trained using pictures from conventional cameras). Last, the viewer orientation should be estimated more precisely using eye-tracking HMD.

ACKNOWLEDGMENTS

This work was partially supported by the Ministry of Science and Technology of Taiwan under the grants No.: 104-2221-E-009-200-MY3, 105-2628-E-001-004-MY2, and 105-2221-E-007-088.

REFERENCES

[1] 2016. Augmented Virtual Reality revenue forecast revised to hit $120 billion by 2020. (2016). https://goo.gl/nw9mtP.

[2] 2016. GLOBAL 360-DEGREE CAMERA MARKET 2016-2020. (2016). https://goo.gl/zJCdnO.

[3] 2016. Next-generation video encoding techniques for 360 video and VR. (2016). https://goo.gl/UNGIwT.

[4] T. Alshawi, Z. Long, and G. AlRegib. 2016. Understanding spatial correlation in eye-fixation maps for visual attention in videos. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME'16)*. 1–6.

[5] A. Borji, M. Cheng, H. Jiang, and J. Li. 2014. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878* (2014).

[6] L. Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. 177–186.

[7] S. Chaabouni, J. Benois-Pineau, and C. Amar. 2016. Transfer learning with deep networks for saliency prediction in natural video. In *Proc. of IEEE International Conference on Image Processing (ICIP'16)*. 1604–1608.

[8] C. Chang, C. Hsu, C. Hsu, and K. Chen. 2016. Performance measurements of virtual reality systems: Quantifying the timing and positioning accuracy. In *Proc. of ACM Conference on Multimedia (MM'16)*. 655–659.

[9] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. 2016. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR'16)*. Cancun, Mexico.

[10] T. El-Ganainy and M. Hefeeda. 2016. Streaming Virtual Reality Content. *arXiv preprint arXiv:1612.08350* (2016).

[11] S. Friston and A. Steed. 2014. Measuring latency in virtual environments. *transactions on visualization and computer graphics* 20, 4 (2014), 616–625.

[12] V. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen. 2016. Tiling in Interactive Panoramic Video: Approaches and Evaluation. *IEEE Transactions on Multimedia* 18, 9 (2016), 1819–1831.

[13] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang. 2015. Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108* (2015).

[14] R. Guntur and W. Ooi. 2012. On tile assignment for region-of-interest video streaming in a wireless LAN. In *Proc. of ACM international workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'12)*. 59–64.

[15] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] Chun-Ying Huang, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu, and Cheng-Hsin Hsu. 2014. GamingAnywhere: The First Open Source Cloud Gaming System. *ACM Transactions on Multimedia Computing, Communications, and Applications* 10, 1 (Jan 2014).

[17] T. Judd, K. Ehinger, F. Durand, and A. Torralba. 2009. Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th international conference on*. IEEE, 2106–2113.

[18] Y. Kavak, E. Erdem, and A. Erdem. 2017. A comparative study for feature integration strategies in dynamic saliency estimation. *Signal Processing: Image Communication* 51 (2017), 13–25.

[19] H. Kimata, D. Ochi, A. Kameda, H. Noto, K. Fukazawa, and A. Kojima. 2012. Mobile and multi-device interactive panorama video distribution system. In *Proc. of IEEE Global Conference on Consumer Electronics (GCCE'12)*. 574–578.

[20] B. Lucas and T. Kanade. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. of the International Joint Conference on Artificial Intelligence*. 674–679.

[21] H. Lakshman M. Yu and B. Girod. 2015. A Framework to Evaluate Omnidirectional Video Coding Schemes. In *2015 IEEE International Symposium on Mixed and Augmented Reality*. 31–36.

[22] A. Mavlankar and B. Girod. 2009. Pre-fetching based on video analysis for interactive region-of-interest streaming of soccer sequences. In

Proc. of IEEE International Conference on Image Processing (ICIP'09). 3061–3064.

[23] A. Mavlankar and B. Girod. 2010. Video streaming with interactive pan/tilt/zoom. In *Signals and Communication Technology*. 431–455.

[24] T. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, and S. Yan. 2013. Static saliency vs. dynamic saliency: a comparative study. In *Proc. of ACM International Conference on Multimedia (MM'13)*. 987–996.

[25] K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[26] K. Skarseth, H. Bjørlo, P. Halvorsen, M. Riegler, and C. Griwodz. 2016. OpenVQ: a video quality assessment toolkit. In *Proc. of ACM International Conference on Multimedia (MM'16), OSSC paper*. 1197–1200.

[27] I. Sodagar. 2011. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia* 18, 4 (2011), 62–67.

[28] E. Vig, M. Dorr, and D. Cox. 2014. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2798–2805.

[29] G. Simon X. Corbillon, A. Devlic and J. Chakareski. 2017. Viewport-Adaptive Navigable 360-Degree Video Delivery. In *Accepted to appear at IEEE International conference on communications (ICCfi17)*.

[30] M. Young, G. Gaylor, S. Andrus, and B. Bodenheimer. 2014. A comparison of two cost-differentiated virtual reality systems for perception and action tasks. In *Proc. of the ACM Symposium on Applied Perception*. 83–90.