Content Assisted Viewport Prediction for Panorammic Video Streaming

Tan Xu
AT&T Labs
1 AT&T Way
tanxu@research.att.com

Feng Qian
University of Minnesota Twin Cities
200 Union Street SE
fengqian@umn.edu

Bo Han
AT&T Labs
1 AT&T Way
bohan@research.att.com

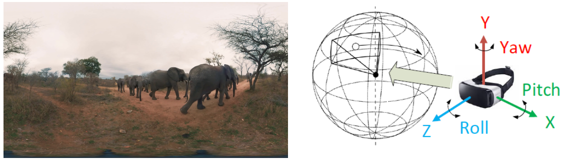
Abstract

In this paper, we explore the viewport prediction problem for 360-degree video streaming by utilizing a viewer's recent head movement trajectory, cross-viewer heatmap, and video saliency detection. We propose a deep neural network (DNN) model using long short-term memory network (LSTM) as its backbone. This model fuses multi-modality features and makes a joint prediction for a user's future viewing direction. We evaluate the proposed approach on a dataset recording the viewing sessions of more than 100 users and show that it outperforms several baseline schemes.

1. Introduction

As a primary Virtual Reality (VR) application, 360° video streaming has become increasingly popular in recent years. For example, in 2017 Facebook 360 witnessed 300 million more 360° video viewers, and nearly 1 million newly uploaded 360° videos[1]; YouTube, another major video streaming platform, can easily filter over 1 million videos with "360" in the title and 360° as video type. This growing popularity is caused by not only the availability of low-price panoramic cameras, but also the immersive viewing experience where viewers can changing their viewing directions freely.

However, it is well known that a 360° video is generally 5-6 times larger than a conventional video under the same visual quality due to its coverage of a larger panoramic scene [4]. Therefore, it requires a high network bandwidth to stream such a video. This issue is getting severer as more demands for higher resolution videos, *e.g.*, 8K, or even 16K. Motivated by this challenge, this study targets on accurate viewport prediction (VP) for 360° video streaming. Prior studies already showed that there is a waste for streaming invisible area in 360° videos. Hence, a high-level idea of viewport guided streaming is proposed, which prefetches only the visible area (called viewport) instead of the entire panoramic scene [5].



(a) Example 360° frame.

(b) 360° video playback

Figure 1: 360° video frame and playback.

Usually, a 360° video is created using an omnidirectional camera or a set of cameras. Then, by using projection, such as Equirectangular projection, the spherical coordinates (longitude and latitude) are forward transformed to planar coordinates in a 2D space, as exemplified in Figure 1a. CubeMap is another commonly used projection method. During playback, as shown in Figure 1b, a player reversely projects each frame from the plane back onto a 3D virtual sphere, and a viewer, such as a user wearing a VR headset, is situated in the center of this virtual sphere. The viewer can freely change her viewing direction, which, together with the device's Field of View (FoV), determine the viewport being rendered on the display.

A device's FoV is usually fixed. Therefore the viewport is mainly determined by a user's viewing direction. Since accurate eye tracking and gaze detection is still not mature, in this study we predict future viewing directions by using head movement traces collected from the device's motion sensors. The direction is measured by longitude and latitude of the virtual sphere centered at the viewer. Our assumption is that a viewer's future viewing direction may be predictable given her recent head movement trajectory, provided video content, and other viewers' viewport trajectories when watching the same 360° video.

In the rest of the paper, we first introduce the dataset in Section 2. We then detail our VP models and features in Section 3, and present evaluation results in Section 4. In Section 5, we conclude this study.

[1] https://blend.media/blog/raindance-film-festival-2017

2. Dataset

In this paper, we use a 360° video dataset created from a user study, which involves 130 participants watching ten popular 360° videos from YouTube by wearing head-mounted display (HMD), a Samsung Gear VR headset [5]. The videos span a wide range of genres including documentary, scenery, movie, performance, diving, driving, skydiving, *etc*. They are all encoded in the standard H.264 format with a 4K resolution, and use the Equirectangular projection. The bitrate of these 360° videos ranges from 12 to 22 Mbps. The videos are limited to be a few minutes to make the data-collection session with a reasonable duration. The participants are diverse in terms of age, gender, and experience of watching 360° videos. In total, the dataset contains 1,300 (130×10) viewing sessions from the subjects and the total duration of the collected trace is 4,420 minutes. Compared to other 360° video datasets, it is the most comprehensive and diverse dataset with the longest viewing time.

Preprocessing. The raw data is collected at 200Hz frequency from the built-in gyroscope sensor using the aircraft coordinate system, which rotates in three dimensions: *pitch*, *yaw* and *roll*. Since viewports mainly change in the vertical and horizontal directions [2], we convert the raw data into the spherical coordinates (*i.e*. latitude and longitude) [2]. We also lower the sampling rate to 30Hz, in order to map the collected data with video frames, which are displayed at the same rate. Note that the horizontal movement can rotate in circles for both left and right directions. For example, when the reported consequent longitude values suddenly change from -170° to +170°, the user actually only moves 20° horizontally in the left direction instead of 340° in the right direction. We handle this issue by adding or subtracting the number of circular rotations (360° for one circulation) for longitude values. This will result in longitude values exceeding the [-180°, +180°] boundaries.

3. Viewport Prediction

In this section, we describe how to extract features from a viewer's recent head movement, provided video content, and other users' viewports when watching the same 360° video, and utilize them to make prediction separately and jointly.

3.1. Trajectory Only Approach

Existing studies showed that VR user's head movement is indeed predictable by examining trajectories [2, 4]. Therefore, in this paper, we first follow a typical adaptive paradigm to predict the viewports in the coming pw (prediction window) seconds by considering recent hw (history window) seconds worth of head movement data. Given its time sequential nature, we propose a Long Short-term Memory (LSTM) network for the task, which can not only capture the long-term dependancies in the data but also avoid the vanishing and exploding gradient problem [1]. Specifically, we implemente our LSTM model using Tensorflow [3]. In the model, we employ 1 layer of LSTM with 64 neurons, with an additional Subtraction layer to conduct point normalization after the input layer, and an Add layer to restore the value back before output. We use ADAM for optimization, and Mean Absolute Error (MAE) as loss function.

For comparison, we investigate several regression methods. They consider time as an independable variable $\mathbf{t}$, and viewing direction $\mathbf{y}$ as the dependable variable. The goal is to learn a regression function $\mathbf{y} = f(\mathbf{t})$ from hw, and predict $\mathbf{y}$ in pw. For these methods, we model latitude and longitude separately. Hence, there are two regression functions, one for each input. Specifically, we train a linear regressor (LR) and a non-linear regressor - MultiLayer Perceptron regressor (MLP). For the MLP, we start with a simple architecture, which contains 1 hidden layer of 3 neurons, and uses hyperbolic tangent function for activation and L-BFGS for optimization. In addition, we use a Static baseline, which takes the last observed viewport in hw for the entire pw.

3.2. Cross-viewer Heatmap

In addition to a viewer's own recent head movement trajectory, other viewers' viewing directions for the same video frame may be suggestive. To justify this assumption, we random select a video in our dataset, and sample 30 viewers' trajectories for the same video. We then plot the latitude and longitude as shown in Figure 3, where the x-axis is the index of video frame extracted at a rate of 30 frames per second (FPS) to align with the HMD data at 30Hz. The black solid line is the median positions cross the sampled viewers, and the range is the mean position with ± 1 standard deviation. From the figure, we can observe a commonality of multiple viewers' head movement traces.

Therefore, given a video frame, we first collect the users' viewing directions (using the original coordinates without longitude correction). Then we project these coordinates ($latitude \in [-90, +90], longitude \in [-180, +180]$) to pixels of a 180×360 image. For each pixel in the image, we count how many times it has been watched, and apply a two-dimensional Gassuian smoothing to the surrounding pixels. The process results in a heatmap for the video frame, as exampled in Figure 2.

3.3. Video Frame Saliency

Given the cross-viewer commonalities when watching the same 360° video, we further assume that it is the content that drives multiple viewers to look at a common area. Therefore, we propose to extract the saliency map for each video frame,

[2] In this study we consider only the *pitch* and *yaw*, as it is known that users rarely change the *roll* (*i.e*. rotating head along the Z axis) when watching VR contents [2].

[3] https://www.tensorflow.org

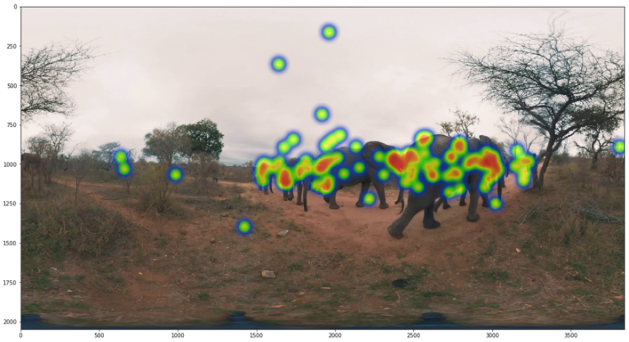


Figure 2: Heatmap.

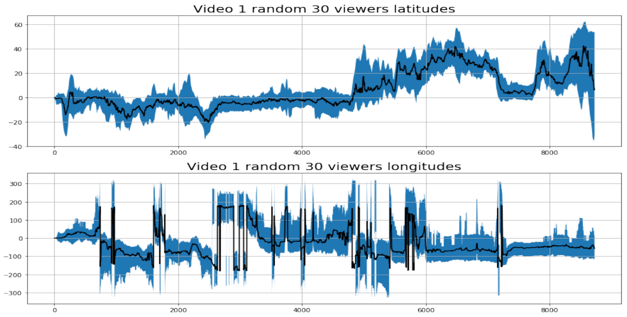


Figure 3: Cross-viewer trajectory commonality.

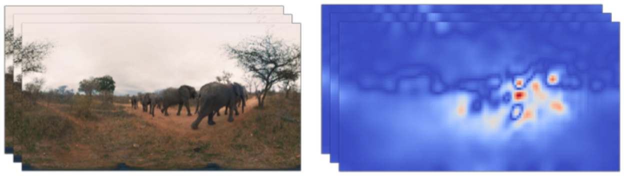


Figure 4: Temporal Ittykoch Saliency Map.

which shall tell us the region of interest and thus can help us predict viewing direction for that frame.

For a particular video frame, in order to extract its saliency map, we apply a classical feature intensive method - Ittykoch, which first decomposes an image into multiple feature channels according to intensity, edge, colors and orientations, and then combines them to identify saliency areas [3]. In addition to the detection of saliency on a static video frame, we further conduct background subtraction to reduce less interested areas. We apply the well known Gaussian mixture-based background/foreground segmentation algorithm for this purpose. The high-level idea is to temporally filter changing pixels between continuous frames [6]. By combining the two processes, we could extract temporal saliency maps for video frames as shown in Figure 4.

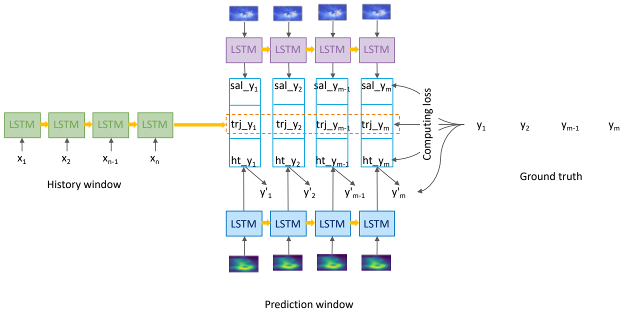


Figure 5: Multi-modality Fusion Model.

3.4. Multi-modality Fusion

After having multiple ways to predict a user's viewing direction, we design a deep learning model as shown in Figure 5 to fuse these modalities. It is composed of three LSTM branches. The trajectory LSTM (green color) takes n coordinates from hw, and predicts m future coordinates in pw, denoted as trj_y_i. The heatmap LSTM (blue color) takes heatmaps of video frames corresponding to each prediction step as input, and outputs a second group of m predictions in pw, denoted as ht_y_i. For each heatmap, we let it go through 3 convolutional layers with a max pooling layer following each. Then, after this image feature extraction, we apply a flatten step and 1 dense layer to regress a coordinate (latitude and longitude). The utilization of LSTM captures the status changes over prediction steps. A similar architecture is also applied to take the saliency maps aligned with the prediction steps, with its outputs denoted as sal_y_i. Finally, we concatenate **trj_y**, **ht_y**, and **sal_y** at each prediction step, and yield one final output **y**.

When training such a model, we choose MAE as the loss function and ADAM as optimization. We examine the losses for not only the final outputs, but also each branch so that their parameters could be better tuned individually and jointly. For both heatmap and saliency-map LSTM branches, we apply TimeDistributed layers so that their parameters are consistent over prediction steps. There are other hyperparameters we choose for the model architecture and training, which we give more details in Section 4.

4. Evaluation

In order to evaluate the performance of our proposed models, we conduct a 2-fold cross validation on the dataset. There are two experiment setup decisions we need to make: (1) The size of pw. We highlight 3 options: $0.1s$, $1.0s$ and $2.0s$, with each we vary hw from 0.05, $0.6s$, and $1.0s$. (2) The number of viewers for training. Despite we split the data into 2 folds, we investigate the impact of the number of training viewers. We select the number of training viewers from $[3, 10, 30]$. We generate heatmap by checking the

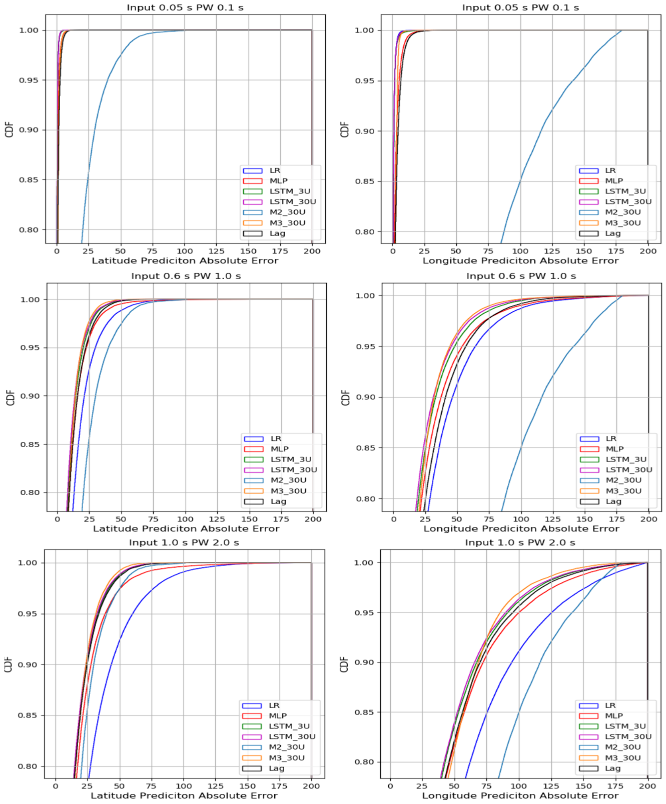


Figure 6: AE CDF comparisons for pw sizes: 0.1s, 1.0s, and 2.0s evaluated on 8 videos.

trajectories from the training viewers for the same video.

Prediction Results. We check Absolute Error (AE) for each prediction of viewing direction compared with the ground truth, the same as computing the MAE loss in model training. Figure 6 shows the cumulative distributions of the AEs for the three pw sizes, with each separately reporting latitude and longitude results. To highlight the differences between models, we only show CDF above 80%.

We have the following observations from the experiment results. (1) For all models the prediction accuracy decreases for longer pw, which indicates long term VP is a more difficult problem to solve. The models can achieve nearly perfect predictions for short pw. (2) For all models, predictions of longitude have around doubled errors than latitude, which is due to the horizontally doubled size of activity area. (3) Regression models can provide accurate predictions only for short pw, and the accuracy decreases faster than other models when pw increases. (4) LSTM based trajectory models consistently outperform baseline models for all pw, but more training viewers does not help improve the accuracy dramatically. (5) Cross-viewer heatmap and saliency map can help with long term VP. They can give a reasonable off-line whole video VP (M2) with consistent performance (independent of pw and no need of hw trajectory inputs), which exceeds some of the trajectory-based models when pw increases. (6) When joining all three modalities (M3), it balances inputs from recent trajectory, cross-viewer interests, and content saliency, which produces optimized predictions for both short and long pw.

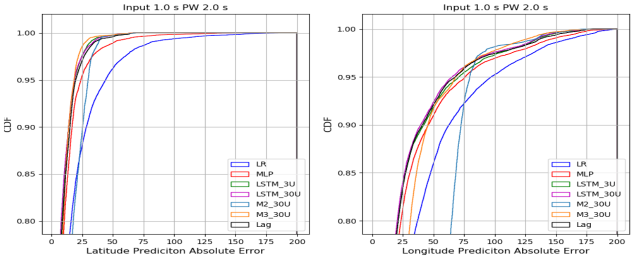


Figure 7: AE CDF comparisons at $pw = 2.0s$ evaluated on 2 videos (Mega.Coaster and GTR.Drives.First.Ever).

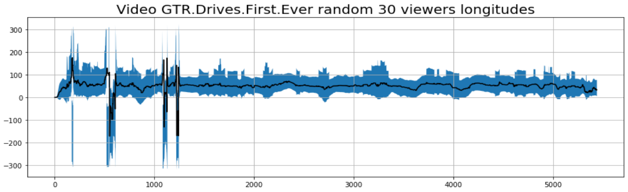


Figure 8: Cross-viewer trajectory for videos GTR.Drives.First.Ever.

However, we observe that the outperformance of M3 in longitude does not apply to two videos (Mega.Coaster and GTR.Drives.First.Ever) as their $pw = 2.0s$ AE CDF shown in Figure 7. After analyzing the data, we notice these two videos are characterized as driving content with high motion content at the side of the driving trails. When watching these videos, the viewports for most of the users are consistently centered around the driving trails. Thus, the audiences are unlikely to change their viewing direction, which results in higher prediction accuracy from trajectory models even at $pw = 2.0s$. The content analysis does not help but may introduce diversions that the audiences may ignore. Figure 8 shows the 30 viewers' longitude trajectories for the GTR.Drives.First.Ever video. We also observe that these two videos have the lowest mean standard deviation of heatmaps compared to other videos.

5. Conclusion

In this study, we explored the problem of viewport prediction for 360° videos. We proposed a DNN model based on LSTM to fuse inputs from a viewer's recent head movement trajectory, cross-viewer heatmap, and video saliency detection, and demonstrated its effectiveness over several baselines and singular modality on a large 360° video dataset.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016. 2

[2] Yanan Bao, Huasen Wu, Tianxiao Zhang, Albara Ah Ramli, and Xin Liu. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In *Proceedings of Big Data 2016*, pages 1161–1170. IEEE, 2016. 2

[3] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10-12):1489–1506, 2000. 3

[4] Feng Qian, Bo Han, Lusheng Ji, and Vijay Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *Proceedings of the Workshop on All Things Cellular: Operations, Applications and Challenges*, pages 1–6. ACM, 2016. 1, 2

[5] Feng Qian, Bo Han, Qingyang Xiao, and Vijay Gopalakrishnan. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proceedings of MobiCom 2018*. ACM, 2018. 1, 2

[6] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004. 3