Your Attention is Unique: Detecting 360-Degree Video Saliency in Head-Mounted Display for Head Movement Prediction

Anh Nguyen
Georgia State University

Zhisheng Yan*
Georgia State University

Klara Nahrstedt
University of Illinois Urbana-Champaign

ABSTRACT

Head movement prediction is the key enabler for the emerging 360-degree videos since it can enhance both streaming and rendering efficiency. To achieve accurate head movement prediction, it becomes imperative to understand user's visual attention on 360-degree videos under head-mounted display (HMD). Despite the rich history of saliency detection research, we observe that traditional models are designed for regular images/videos fixed at a single viewport and would introduce problems such as central bias and multi-object confusion when applied to the multi-viewport 360-degree videos switched by user interaction. To fill in this gap, this paper shifts the traditional single-viewport saliency models that have been extensively studied for decades to a fresh panoramic saliency detection specifically tailored for 360-degree videos, and thus maximally enhances the head movement prediction performance. The proposed head movement prediction framework is empowered by a newly created dataset for 360-degree video saliency, a panoramic saliency detection model and an integration of saliency and head tracking history for the ultimate head movement prediction. Experimental results demonstrate the measurable gain of both the proposed panoramic saliency detection and head movement prediction over traditional models for regular images/videos.

CCS CONCEPTS

• **Information systems → Mobile information processing systems**;

KEYWORDS

360-degree video; head movement prediction; saliency

ACM Reference Format:
Anh Nguyen, Zhisheng Yan, and Klara Nahrstedt. 2018. Your Attention is Unique: Detecting 360-Degree Video Saliency in Head-Mounted Display for Head Movement Prediction. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240669

*Z. Yan is the corresponding author. zyan@gsu.edu

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240669

1 INTRODUCTION

With the annual growth rate of head-mounted display (HMD) and 360-degree cameras reaching an impressive 56% [8] and 35% [33], respectively, 360-degree videos are becoming more popular than ever before. Major video websites such as YouTube, Facebook, and Vimeo, have all been promoting their 360-degree video services aggressively. Although basic 360-degree video service of mediocre quality is currently available, streaming and rendering the ultra-high-resolution (up to 16K) panoramic videos with a negligible delay for real-life immersive and interactive experience is still an open problem that is yet to be resolved.

Among various research efforts towards the desired experience, *head movement prediction* is one essential yet daunting task that needs to be addressed urgently. Accurate head movement prediction can enable bandwidth-efficient 360-degree video streaming [9, 22, 28], where the client only downloads the video portions that the user is likely to view in high quality while the remaining portions are ignored [38, 39] or fetched in low quality [5]. Furthermore, predicting head movement accurately can significantly reduce the motion-to-photon delay [35] since it would be possible to render exactly one viewport from the downloaded video portions in advance before the head moves.

It is well known that understanding users' visual attention in HMD is the key to head movement prediction since users are more likely to stay on regions of interest. Proper video *saliency detection* can thus strongly benefit the head movement prediction. By combining saliency-based head movement prediction with head orientation history [10], an even more accurate head movement prediction can be achieved.

Despite this general consensus, we observe that our understanding of saliency detection for 360-degree videos is still limited. The reason behind this observation is that 360-degree videos introduce new effects on the visual attention of HMD users while little is known regarding the true saliency under this new context. First, prior saliency detection models are customized for regular videos with central bias [23, 32] that places important details on video frame center. They cannot be directly applied to 360-degree videos where user attention may spread out the entire equator. Second, for a typical multi-object 360-degree video, traditional saliency detection schemes would treat the video as a single viewport and identify front objects as more salient objects. However, there is no such preference during a user's multi-viewport viewing since objects in the front and back of an equirectangular frame can be in separate viewports and the user can focus on whichever objects that fit into her actual viewport.

In this paper, we bridge the aforementioned gap by detecting *panoramic saliency* that captures user's unique visual attention on 360-degree videos, which shifts traditional saliency detection that

has been studied for decades to a new saliency detection specifically tailored for 360-degree videos. By leveraging true 360-degree video saliency, we aim at ultimately enabling far more accurate head movement prediction in HMD. To achieve this objective, we are facing several research challenges.

- *Building a dataset for panoramic saliency.* There is no public dataset that labels the saliency of 360-degree videos. With the limited head tracking data available publicly, it is non-trivial to construct an appropriate dataset for panoramic saliency.
- *Training a saliency detection model for 360-degree videos.* The relationship between the content features and saliency in 360-degree videos is not yet clear. Inappropriate feature extraction and model prediction would lead to unacceptable saliency detection performance.
- *Consolidating a head movement prediction model for 360-degree videos.* Head movement in HMD is not purely determined by saliency. How to combine detected 360-degree video saliency with other types of data, e.g., motion data, to ensure the overall performance is critically important.

To tackle these challenges, we start with constructing a completely new dataset for panoramic saliency from two existing head tracking datasets. By mapping head orientation logs into user fixation, we derive the panoramic saliency of 360-degree videos based on inter-agreement among all users' fixation. Furthermore, considering the lack of huge amount of data for training from scratch, we employ transfer learning on a traditional saliency model to obtain *PanoSalNet*, the proposed panoramic saliency detection model that leverages the benefit of deep neural networks to extract novel features for 360-degree videos automatically. Finally, we integrate the saliency map and head orientation log, and utilize Long Short-Term Memory (LSTM) model to exploit the interplay between head orientation and panoramic saliency at different time moments in the past to predict future head orientation. We validate the proposed datasets and models by extensive evaluations. Results show that PanoSalNet and the proposed head movement prediction significantly outperform traditional models for regular images/videos.

To summarize, the contributions of this paper include:

- A public dataset of panoramic saliency for 360-degree video studies (Section 3).
- An accurate panoramic saliency detection model to identify the visual attention in 360-degree videos (Section 4).
- A promising head movement prediction model to allow various system designs in 360-degree videos (Section 5).

2 RELATED WORK AND MOTIVATION

2.1 Saliency Detection

Saliency detection has long been an important area for predicting visual attention in image/video viewing. Early works focused on studying various hand-crafted features to improve the detection accuracy [11, 13, 23]. Others add a preprocessing step based on low level features to improve performance [26] or reduce distortion [24, 25]. With the superior performance of Convolutional Neural Network (CNN), extensive efforts have been made towards CNN-based saliency models. DeepGaze [19] adapted a CNN for image classification to saliency prediction using transfer learning. Pan *et al.* trained a shallow network for saliency detection from scratch and applied transfer learning to generate a deep network [27]. Zhang *et al.* used shallow learning to predict saliency specifically for videos with crowd scenes [40]. These models that are designed for regular images and videos under smartphone/computer viewing build the foundation of saliency detection, but they are not directly applicable to 360-degree videos under HMD viewing.

Saliency detection for 360-degree images has been recently introduced in limited studies. In general, saliency maps for 360-degree images are first collected when the users freely interact with the images under HMD. Then the 360-degree image saliency is utilized to translate traditional image saliency by linear weighting [1, 31] or to adapt a traditional saliency model to a new model for 360-degree images by transfer learning [21]. These 360-degree image solutions provide insights for saliency study in HMD. However, during saliency collection, image content is *static* and users are allowed to go back and forth along the 360-degree image for as many times as possible to "find" the salient objects. Such a viewing is in sharp contrast to 360-degree video viewing where the content is *dynamic* and users may easily miss objects when moving the head around. Therefore, 360-degree image saliency cannot accurately reflect the saliency in 360-degree video viewing.

Unlike the aforementioned models, we instead probe into the new space of panoramic saliency for dynamic video content by exploring the unique visual attention in 360-degree videos. We aim at accurately pinpointing the salient regions of 360-degree videos.

2.2 Head Movement Prediction

Head movement prediction has been an indispensable component in 360-degree video streaming systems. Most existing systems conducted a basic processing of head movement history to predict the future movement, such as simple average [5], linear regression [9, 39], and weighted linear regression [22, 28].

Research efforts dedicated to head movement prediction algorithms for 360-degree videos are limited. Aladagli *et al.* predicted head movement based on a pre-trained saliency model [2]. Fan *et al.* integrated saliency and prior head orientation to further improve the prediction accuracy [10]. However, both schemes rely on traditional saliency models for regular videos and may not capture the unique visual attention in 360-degree videos. Scan path prediction for 360-degree images was investigated in [3]. Unfortunately, due to the distinct viewing behavior for dynamic video content, scan path of a static image is unlikely to match the head movement under 360-degree videos.

In this paper, we remedy these issues to take a step further. We exploit the proposed panoramic saliency crafted for dynamic 360-degree videos along with user's head movement history, as well as the interplay between them, to maximize the head movement prediction performance.

2.3 Motivation

In this section, we elaborate on the necessity of a customized saliency model for head movement prediction in 360-degree videos by identifying two intrinsic problems of traditional saliency models for regular images/videos.

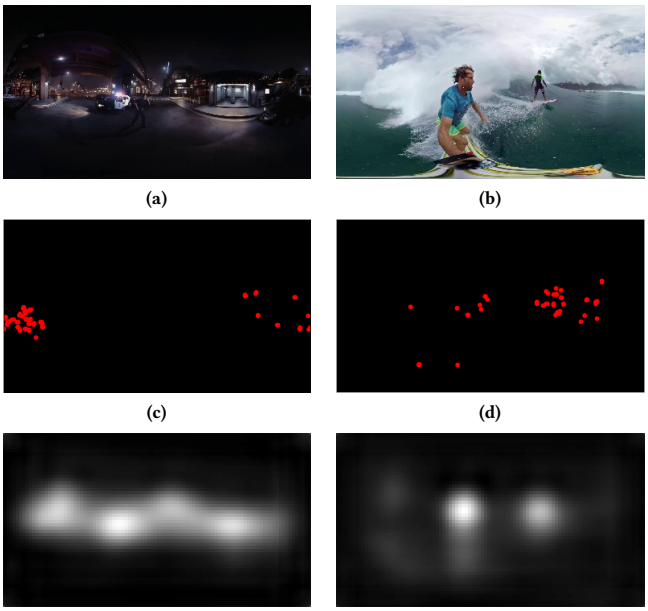

Figure 1: Saliency detected by traditional models does not match true user fixation. a) and b): sample frame; c) and d): true user fixation. e) and f): salient prediction result from a traditional model [27].

Central Bias. In regular videos with a single viewport, salient objects are normally found at the frame center. As a result, trained models from these saliency datasets of regular videos tend to have a central bias [23, 32], where the level of saliency is reduced as the content moves from frame center to the four edges. However, the central bias would not reflect the saliency of 360-degree videos. In a typical equirectangular frame, although objects at the two poles (top and bottom) are rarely viewed by users, all objects along the equator may attract user attention. In other words, edge objects can also be the salient objects in some 360-degree videos. As shown in the example of the left column in Figure 1, users are more interested in the small animal at the edge of equator while the central biased saliency detected by a traditional model [27] is completely different from the true user fixation.

Multi-object Confusion. During the saliency data collection for regular videos, users are able to quickly scan through all objects in the single viewport with a limited field of view and are generally more interested in front objects than objects in the back. The resulting saliency model adapts to this behavior and detects the saliency accordingly. Unfortunately, such models may generate saliency significantly deviating from true user fixation when applied to an equirectangular frame. For multi-viewport 360-degree videos, a front object and a back object in an equirectangular frame can lie in two separate viewports in the 3D viewing space. A user may focus on any of them as long as the object is within her actual viewport, indicating that the front object in the equirectangular frame does not necessarily obtain more attention. The right column of Figure 1 illustrates this phenomenon. The persons in the equirectangular frame are 90-degree apart from each other in two viewports. The majority of users pay attention to the person further away since the closer man is holding the 360-degree camera and the viewport including him is occupied by his body without interesting views. Unfortunately, the traditional model is ignorant of this and still assigns a higher saliency to the front person.

Building from the above analysis, we conclude that it is imperative to develop a unique saliency detection model for 360-degree videos, which will consider all objects along the equator and pinpoint the most interesting object among multiple objects in equirectangular frames. Such accurate saliency detection would naturally benefit the head movement prediction.

3 DATASET

Since there is no existing saliency dataset specifically for 360-degree videos, we have created a new dataset. In this section, we start with describing the steps to generate our dataset. We then carefully examine the dataset to demonstrate that the created panoramic saliency data is highly consistent with human viewing fixation.

3.1 Collection of Panoramic Saliency

To collect the saliency maps of 360-degree videos, it is essential to extract user fixation. The fixation points imply the region that users pay special attention. In regular image/video saliency dataset, eye gaze points are obtained by specialized eye-tracking devices to derive fixation. Due to the absence of eye tracker in HMD, we adopt a similar method as in [1, 34] to represent eye gaze point by head orientation. This methodology is supported by the fact that the head tends to follow eye movement to preserve the eye-resting position (i.e., eyes looking straight ahead) [17]. We now follow the similar procedure in prior saliency collection works [15, 34] to extract fixation and generate the panoramic saliency dataset.

Deriving Head Orientation. To collect head orientation data, we explore two public head movement datasets for 360-degree videos [6, 37]. The first dataset [37] has 18 videos viewed by 48 users in 2 experiments. We select the 9 videos in the first experiment where the head orientation is obtained during free viewing without any particular viewing task. The second dataset [6] includes five videos freely viewed by 59 users. We choose two videos from the dataset because the fixation points of the other three videos are noisy, implying no region of interest. Both datasets record timestamped head orientation and the corresponding frame under viewing. A head orientation sample is stored as a quaternion, a four-tuple mathematical representation of head orientation with respect to a fixed reference point. We convert the quaternion to a regular 3D unit vector v ($|v| = 1$) to represent the head orientation [18]. Coupled with the timestamps, we are able to derive where the user is looking at on the 3D sphere for any given moment.

Extracting Fixation. We then process the head orientation (or equivalently gaze point) to extract the fixation. Fixation occurs when user head orientation fixates at a specific region for a short period of time. Note that fixation would not be found at every time step since there are cases called *saccade* where the head quickly moves from one interesting region to another. We derive fixation by first removing saccade (fast head orientation change) from the data. Based on head movement velocity and acceleration derived from timestamped head orientation, the saccade can be identified using the threshold-based method suggested by [12]. In particular,

Table 1: Panoramic Saliency Dataset Evaluation

Model	sAUC	NSS	CC
Dataset Saliency	0.7966	1.9864	0.2521
Equator Bar	0.5012	0.8086	0.1078
Circle at Center	0.4462	0.3424	0.0487

all head movement with velocity over $20°/s$ and acceleration over $50°/s^2$ is considered as a saccade. We then associate the filtered head orientation logs with the video frame under viewing, and map a head orientation v in the 3D sphere to a fixation point (a pixel) in an equirectangular frame by

$$a = \frac{\phi}{360} * \mathcal{W} \tag{1}$$

$$b = (\frac{1 - \sin(\theta)}{2}) * \mathcal{H} \tag{2}$$

where a and b are the longitude and latitude positions in the equirectangular frame, ϕ and θ are the vertical and horizontal angles of v in 3D space, and $\mathcal{W}$ and $\mathcal{H}$ are the width and height of the target equirectangular frame. We herein focus on the equirectangular planar frame as it is the most common format of 360-degree videos.

Creating Fixation Maps. After the fixation points on corresponding equirectangular frames are identified, we produce the fixation map, which is a collection of fixation from multiple users for a frame. To remove noisy points far away from the fixation, we further apply Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm over equirectangular fixation map. This method has been used when processing fixation on 360-degree images [1]. DBSCAN enjoys the advantage that it does not require a predefined number of groups like K-means and does not introduce new points into the dataset.

Outputting Saliency Maps. Fixation points from multiple users in a specific region are usually insufficient to depict a salient region. This is because fixation points are discrete and thus the fixation of different users are unlikely to match exactly. We therefore apply Gaussian Filter to generalize and smooth these scattered user fixation points to a statistical region [15, 34]. This classic method eventually generates the panoramic saliency map.

We repeat the aforementioned procedure to the selected 11 videos and generate 7,000 equirectangular frames with a fixation map and a saliency map. We will use this new dataset for 360-degree videos to train and validate the panoramic saliency detection.

3.2 Dataset Evaluation

To justify the proposed saliency dataset for 360-degree videos, it is important to evaluate whether or not the generated saliency is consistent with the human fixation.

We compute multiple correspondence measures [4] between the saliency generated in the proposed dataset and the user fixation, i.e., shuffle Area Under Curve (sAUC), Normalized Scanpath Saliency (NSS), and Pearson's Correlation Coefficient (CC). These are popular metrics that have been used and discussed extensively in previous works [4, 7, 20, 23]. We also compute the correlation between user fixation and two baselines that simulate equator-biased and central-biased saliency, respectively. Equator Bar is a model where the saliency is linearly decreased from 1.0 to zero when the latitude is varied from 0° (equator line) to ±90° (two poles).

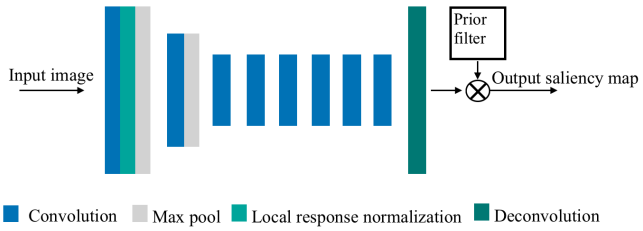

Figure 2: Deep neural network architecture of PanoSalNet.

Due to the projection to equirectangular format, the saliency will nonlinearly decrease from equator line to top and bottom line in a frame. Similarly, the second baseline, denoted by Circle at Center, labels the frame center point as the highest saliency (1.0) and gradually decreases the saliency by expanding a circle around the center point.

The dataset evaluation results are shown in Table 1. The proposed dataset achieves significantly higher values for all metrics (higher correlation to fixation) than two heuristic baselines. This level of dataset performance is consistent with state-of-art saliency dataset in [27], indicating that we have created a reasonable dataset. Furthermore, we can conclude that panoramic saliency cannot be easily obtained by simple central-biased or equator-biased prediction. Instead, a more sophisticated model is certainly needed. Finally, since Equator Bar outperforms Circle at Center, we confirm that users prefer the equator more than the center of equirectangular frames during 360-degree video viewing.

4 360-DEGREE VIDEO SALIENCY DETECTION

In this section, we propose PanoSalNet, the panoramic saliency detection model for 360-degree videos using Deep Convolutional Neural Network (DCNN). In the following, we introduce the network architecture and model training in details.

4.1 Network Architecture

In general, a larger amount of data is needed to avoid over-fitted DCNN model and to produce a better DCNN performance. Unfortunately, since head movement prediction is still at its infant stage, there is only a limited number of head tracking datasets can be used for creating panoramic saliency. Considering the nature of subjective tests, it is even unlikely in the future, if not impossible, to collect millions of saliency maps as in image classification. To address this issue, we propose to employ transfer learning to adapt an existing model to the target DCNN for panoramic saliency detection. Transfer learning is a popular technique to bring pre-trained deep learning models into other expert domains that significantly reduces the amount of required training data. This technique has been successfully applied to problems with very small domain datasets [20, 36]. A similar strategy has also been adopted in the training of saliency detection for 360-degree images [3, 21].

The network architecture of PanoSalNet is illustrated in Figure 2. The proposed network architecture with nine (de)convolution layers is inspired by Deep Convnet, a state-of-art DCNN for saliency detection of regular images. The first three layers enjoy the same structure of VGGNet[30], which allows us to initialize the parameters of these layers by a popular deep learning network that has

shown outstanding performance on image classification task. The next five layers are initialized and trained from scratch on SALICON [14], a saliency map dataset for regular images. To apply transfer learning on such a network, the last few fully connected layers, which contain most of the network parameters, are usually removed and replaced by layers suitable for the requirement of the new application domain [27]. However, since this architecture from Deep Convnet has no fully connected layer, we conduct transfer learning over all layers after the initial parameters of these layers are obtained as mentioned above. This way, we adapt this traditional model to the proposed PanoSalNet.

The predicted saliency will be enhanced one more time as the final output by a prior filter [7], which lowers the saliency in areas based on a priori knowledge, such as four corners of an equirectangular frame.

4.2 Model Training

Investigation on our saliency dataset shows that fixation points tightly cluster around the region of interest when they appear initially. Then they will scatter around the original clustering point when some users move their head away. We have found that such scattered fixation points would prevent the model learning meaningful patterns and significantly degrade detection performance. Thus, to increase the quality of input saliency maps for model training, we select 400 pairs of video frames and saliency maps from our panoramic saliency dataset. This training data is selected such that fixation points concentrated. We also guarantee that there is not too many similar frames for the same video scene, which avoids over-fitting on a few video scenes having a large number of frames. Note that the amount of data is sufficient and is consistent with previous models using transfer learning that have 40-1000 frames of domain data [3, 20, 36].

To expedite the model learning, the frame resolution is downscaled to 512×288. The data are also normalized to (-1, 1) so that data samples center around the zero point. Since predicting saliency map is a regression problem, we use Euclidean distance to measure the difference between ground truth saliency maps and the predicted results. The loss function is defined as follows,

$$L = \frac{1}{N}\sum_{i} L_i(f(X_i, W), y_i) + \lambda R(W) \tag{3}$$

where the batch size N is set at 3, W is a model parameter to be learned, L_i is the Euclidean distance between the output saliency and the ground truth saliency y_i, and $R(W)$ is the regularization expression. PanoSalNet uses standard l_2 for regularization to control over-fitting with weight decay multiplier $\lambda = 5e-4$. The f function calculates the output saliency map based on the input image X_i. The proposed model is trained using Stochastic Gradient Descent (SGD) with momentum. SGD is a popular method to update model parameters to reduce the value of loss function L. For each iteration, model parameters are updated using the following rules,

$$\begin{aligned} u_{t+1} &= \rho u_t + \alpha \nabla L(W_t) \\ W_{t+1} &= W_t - u_{t+1} \end{aligned} \tag{4}$$

The momentum u_{t+1} accumulates gradient values to speed up the learning. It is controlled by parameter $\rho = 0.9$. The network is trained with fixed learning rate $\alpha = 5e-9$. PanoSalNet is tested

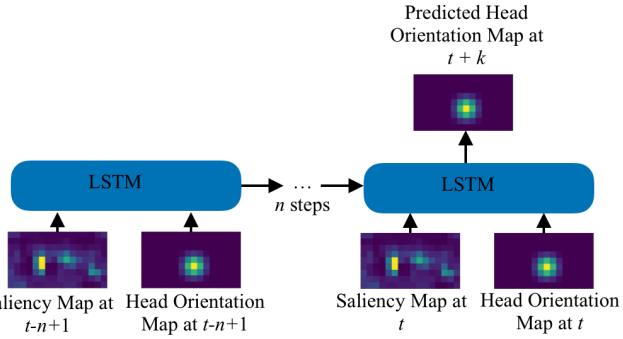

Figure 3: The proposed LSTM architecture.

every 100 iterations to check training progress. We stop the training at 800 iterations to prevent over-fitting.

After model training, the proposed PanoSalNet is then able to predict a saliency map of panoramic saliency based on the input 360-degree video.

5 HEAD MOVEMENT PREDICTION IN HMD

One key challenge of head movement prediction is when the user takes a fast head movement, during which the prediction accuracy has been observed to drop significantly [28]. This type of fast head movement can occur when a new object is presented. Since a user typically moves her head to the most salient region of the video scene, a more accurate saliency detection can potentially address the fast movement prediction and improve the accuracy of head movement prediction. To this end, we integrate the panoramic saliency maps generated from PanoSalNet with user head orientation history for head movement prediction. We exploit both factors and their interplay to maximize the prediction accuracy.

5.1 Model Architecture

In order to learn the pattern of head tracking logs of multiple users and to capture the interplay between temporal user behavior and multiple saliency maps from the past video frames, a highly nonlinear learning model is required. We propose to utilize Long Short-Term Memory (LSTM) model since LSTM is proven to be able to handle a large amount of temporal data and outperform other similar algorithms [29].

In particular, the proposed LSTM model is a Recurrent Neural Network (RNN) that works on the temporal domain. Since stacking multiple layers of LSTM on top of each other can handle more complex data [29], we adopt an LSTM network with the model hyperparameters of α layers and β neurons per layer for the model training and validation. The network architecture of the head movement prediction is shown in Figure 3. The head movement prediction network receives input features from a given number of previous time steps, and provides the prediction of head orientation in the next k time steps (prediction window). The input features of the LSTM network include both the panoramic saliency map detected by the proposed PanoSalNet (indicating regions of interest) and the head orientation feature (recording head movement history).

To better correlate head orientation with requested/viewed video tiles in 360-degree video streaming, we follow a similar method

Table 2: Panoramic Saliency Detection Performance

Model	sAUC	NSS	CC
Dataset Saliency	0.7966	2.4806	0.2885
Deep Convnet	0.6320	1.3256	0.1982
PanoSalNet	0.7112	1.9864	0.2521

in [10] and represent the head orientation feature by *head orientation map*, a spatial data structure similar to saliency map. Head orientation map highlights the viewport within a frame that would be viewed under current head orientation. We generate the head orientation map by first identifying the tile pointed by current head orientation vector and set its likelihood to be viewed as 1.0. Using this tile as the center, we then apply a Gaussian kernel to gradually select other tiles with a lower likelihood to be viewed around the center tile until the selected tiles can cover a viewport.

5.2 Model Training

We use 5 videos from our dataset for model training and another 4 videos for model validation. For each video, we select one segment with a length of 20-45 seconds. The video segment is selected such that there are one or more events in the video that introduce new salient regions (usually when new video scene is shown) and lead to fast head movement of users. We extract the timestamped saliency maps and head orientation maps from these videos, generating a total of 300,000 data samples from 432 time series using viewing logs of 48 users.

Before the model training, we normalize the values in saliency maps and head orientation maps to (-1, 1). The loss function is calculated based on the Euclidean distance between predicted head orientation map and ground truth head orientation map. Our model parameters are updated with Root Mean Square Propagation (RMSprop) method. RMSprop can dynamically adjust the learning rate during training time. It is a preferred method for RNN as suggested by [16].

6 EVALUATIONS

In this section, we evaluate the performance of the proposed saliency detection model PanoSalNet and the head movement prediction model by comparing them with existing models. We implement both models using Intel AI DevCloud that has pre-installed popular machine learning frameworks. Specifically, PanoSalNet is implemented under the provided Caffe framework while the head movement prediction model is implemented on Keras using Tensorflow as the backend. Both experiments are written in Python language. The code is submitted to Intel AI DevCloud as batch jobs. We choose one Intel Xeon processor with 24 cores for both experiments.

6.1 Saliency Detection Evaluation

We evaluate the saliency detection performance by comparing the proposed PanoSalNet with Deep Convnet [27], a state-of-art saliency model for regular images/videos. It achieves competitive results on popular saliency dataset such as iSUN and MIT300. Its weaker variant, the ShallowNet, is the winner of the 2015 LSUN challenge[23]. Therefore, Deep ConvNet is a strong and timely baseline to evaluate our panoramic saliency dataset. We randomly select

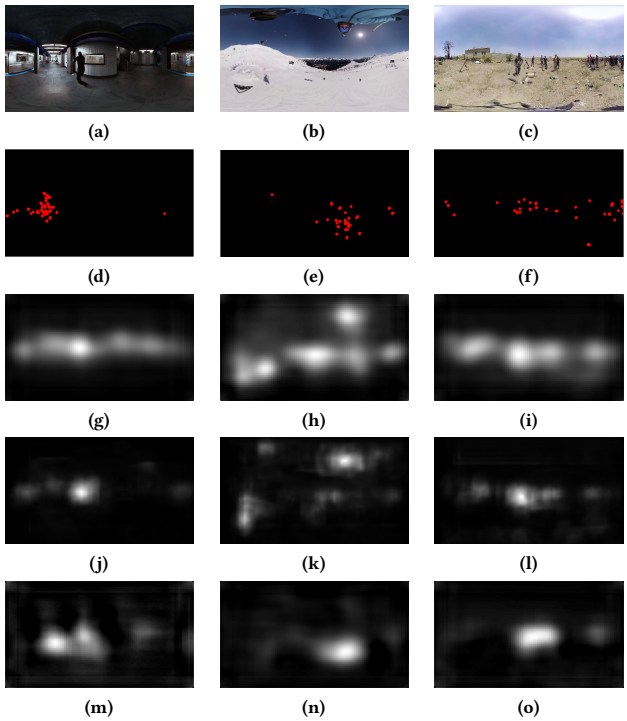

Figure 4: User fixation and saliency detection results of various models for example videos. a)-c) original frame; d)-f) fixation map; g)-i) saliency map of Deep Convnet; j)-l) saliency map of ML-Net; m)-o) saliency of PanoSalNet.

1000 frames from the dataset in Section 3 and pass each equirectangular frame into PanoSalNet and Deep Convnet to produce the predicted saliency. We also use the ground truth saliency collected in the dataset to obtain performance upper bound (denoted by Dataset Saliency).

6.1.1 Saliency Performance Metrics. We first evaluate the saliency detection performance using the three metrics mentioned in Section 3 (the higher the better), which measure the correlation between the generated saliency map and true user fixation. As shown in Table 2, PanoSalNet performs reasonably well compared to the upper bound of Dataset Saliency and significantly outperforms Deep Convnet in all metrics. PanoSalNet learns the unique visual attention of 360-degree viewing and largely boosts the correlation between predicted saliency and true user fixation. On the other hand, Deep Convnet is designed for regular images/videos and suffers the problems of central bias and multi-object confusion, and hence introduces unsuitable saliency for equirectangular frames.

6.1.2 Illustrative Examples. We show the saliency detection results of some example video frames for different saliency detection models. The benchmark models include Deep Convnet [27] and the saliency model in [7] (denoted by ML-Net).

Figure 4 shows three original video frames and their true user fixation, as well as the predicted saliency obtained by the saliency models for comparison. The results show that PanoSalNet has learned some intrinsic patterns from 360-degree videos. For example, it does

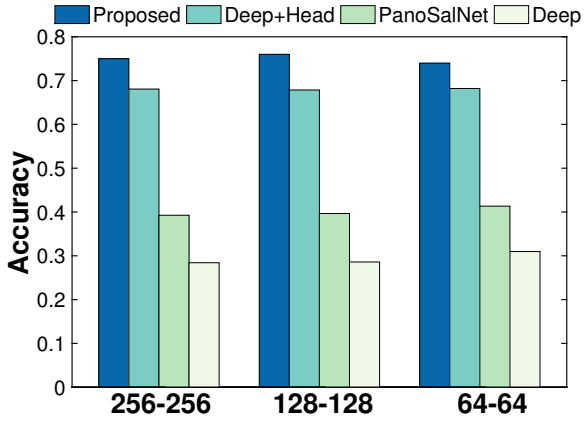

Figure 5: The accuracy of the proposed head movement prediction model outperforms the benchmarks under different training settings.

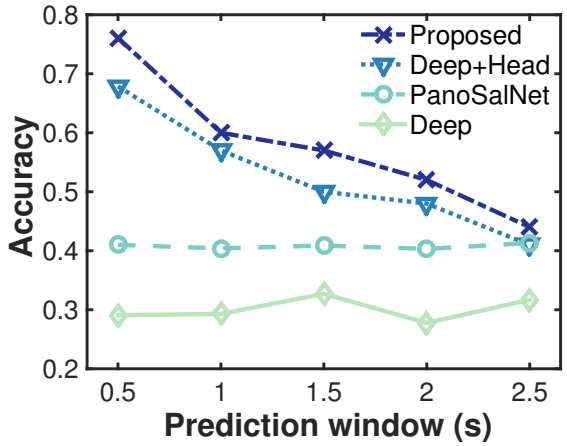

Figure 6: The impacts of prediction window on the accuracy of the head movement prediction models.

not incur central bias and can address the multi-object confusion in equirectangular frames effectively.

In particular, in the first frame, user fixation is on the small animal in the corridor. PanoSalNet predicts the corridor correctly and places a less weight on the policeman. Deep Convnet also includes the animal but since it covers a lot of regions, leading to a lower precision score. ML-Net labels the policeman with the highest saliency without the small animal. In the second frame, PanoSalNet learns the region at the slope, which is typically the target in moving videos. Unfortunately, Deep Convnet and ML-Net ignore the equator bias, thus mark all visible areas around the frame center, including the twisted person at the top edge. This does not match 360-degree video viewing behavior because users in HMD environment have to actually look all the way up to see this person and they would rarely move the head as this [1]. The third image is a difficult case for saliency detection since users are split into two groups. Some users concentrate on the men at the frame center while some other users choose to focus on the group at the right edge. PanoSalNet accurately identifies the center group and assigns a lower saliency to the group at the right edge. On the other hand, ML-Net and Deep Convnet both mark more objects than desired, including the house on the left back and the open region between these objects.

6.2 Head Movement Prediction Evaluation

We now evaluate the proposed LSTM head movement prediction using both head orientation map and the saliency map detected by PanoSalNet as the input features. We compare its performance against several benchmarks by varying the input features of the LSTM model, i.e., (a) using Deep Convnet predicted saliency maps only (denoted by *Deep*), (b) using PanoSalNet saliency only (denoted by *PanoSalNet*), and (c) using both Deep Convnet saliency maps and head orientation maps (denoted by *Deep+Head*).

The evaluation metric is the accuracy of head movement prediction. Accuracy [10] is calculated based on the ratio of the number of overlapping tiles between predicted and ground truth head orientation map over the total number of predicted and viewed tiles. The model is trained on Keras framework for over 40,000 iterations. For every 4,000 iterations, the average accuracy over all frames of the validation set is recorded. We identify the best accuracy that a model can get through increasing iterations and report the result for each model.

Note that the validation set used in this evaluation was never used to train the LSTM model. This can evaluate if it is feasible for the proposed LSTM model to adapt to novel data. We use the input feature from the past one second to predict the head orientation in the future. The equirectangular frame is spatially configured into 16×9 tiles. We also vary the number of layers α and the number of neurons β and represent different cases by LSTM α-β. The default prediction window k is set to be 0.5 seconds.

6.2.1 Accuracy. The prediction accuracy of the various models under a different number of layers and neurons is shown in Figure 5. It can be seen from the figure that the proposed head movement prediction significantly outperforms the benchmarks. On average, the proposed model achieves an accuracy that is 1.9 times over PanoSalNet, 2.6 times over Deep, and 9% higher than Deep+Head. The enhanced performance is attributed to the fact that the proposed prediction model leverages the unique panoramic saliency during 360-degree video viewing as well as absorbing the interplay between panoramic saliency and the head orientation history. On the other hand, Deep+Head only treats the 360-degree video as regular video and thus the predicted visual attention is not accurate. Furthermore, from the degraded performance of PanoSalNet and Deep, we can also infer that head orientation and its temporal interplay with saliency is as important as panoramic saliency in predicting head movement. Finally, since a significant accuracy improvement is not observed when increasing the number of layer and neurons in the prediction models, we conclude that a complicated and deeper model may not be needed, especially under resource-constrained condition.

6.2.2 Impacts of Prediction Window. To explore the effect of prediction window k on the accuracy of the proposed model and other three benchmarks, we vary k from 0.5 seconds to 2.5 seconds.

The results are shown in Figure 6. Thanks to the advantages of identifying panoramic saliency and integrating head orientation, the proposed model achieves a higher accuracy consistently.

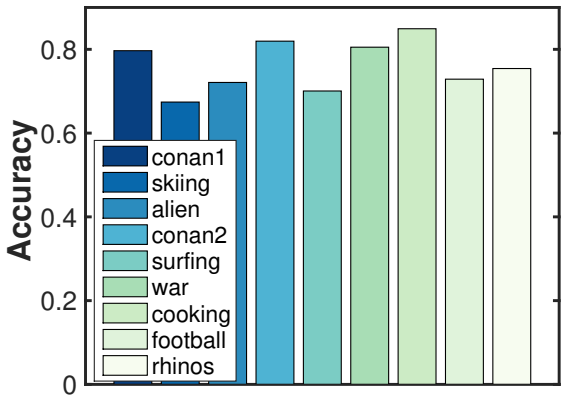

Figure 7: The accuracy of the proposed head movement prediction model under different videos.

Furthermore, we observe that the accuracy of the proposed model and Deep+Head drops at a similar rate as the prediction window enlarges and begins to converge at 2.5 second window size. This is because the head orientation history becomes less relevant when predicting the further future. At 2.5 second, prediction becomes difficult even with the combination of head position and saliency. However, an interesting observation is that models using saliency maps as input features, i.e., PanoSalnet and Deep, show a relatively stable accuracy as k increases. In fact, we verify that the accuracy of PanoSalnet and Deep remains at a similar level at 0.34 and 0.29 even when the prediction window increases to 6.3 seconds. This may be because the temporal correlation between consecutive saliency maps is much less than that between consecutive head orientation maps. Without connecting and interplaying with head orientation history, temporal information provided by consecutive saliency maps drops to a minimal, approximating saliency map as a timely independent feature. Based on the resilience of saliency map to prediction window size, we believe that saliency map may play an important role in predicting head movement in the far future.

6.2.3 Impacts of Video Content. To understand how the proposed model would adapt to different video content for head movement prediction, we use the proposed model to predict head movement on 9 videos from [37]. The nine videos can be divided into three categories: static scenes with a few scene switches (conan1, conan2, war, football, cooking, rhinos), fast-moving scenes (skiing, surfing) and slow-moving scenes (alien). We re-train the proposed model using the data of 40 users and the rest data of 8 users is used as novel data for validation.

We show the accuracy of each video in Figure 7. It can be seen that the proposed model achieves the best performance on static videos, e.g., reaching an accuracy of 85% on "cooking" that has one static scene. On the other hand, a decreased accuracy is observed for fast moving videos such as skiing. This result could be due to the fact that content trajectory is not considered in the model, making it slow or difficult to capture the visual attention change. We will discuss this issue and potential future work in Section 7.

7 DISCUSSION AND FUTURE WORK

Panoramic Saliency Dataset. Although PanoSalNet effectively learns important viewing pattern for 360-degree videos, there might be other inherent attention that is not yet discovered. As the first attempt, the created dataset for panoramic saliency may not include all unique viewing patterns in 360-degree videos. However, with the emergence of 360-degree video research, a larger dataset for panoramic saliency can be collected based on the methodology in Section 3 to improve the performance of saliency detection model.

Content Trajectory Feature. Our experiments and previous studies [10] have shown that content trajectory of moving objects impacts head movement prediction in 360-degree videos. However, projection from the sphere to the equirectangular frame would cause the distortion of content trajectory, e.g., a tiny content motion can be stretched out to a long trajectory. A separate study to address the distortion and the modeling of content trajectory in 360-degree videos is definitely needed to replace traditional content trajectory based model. Once this feature space is better understood, we expect to achieve an even higher prediction accuracy.

System Integration. Integrating the proposed head movement prediction framework into 360-degree video systems requires further treatment of two issues. First, we point out that the time overhead of running the head movement prediction is minimal since the training of LSTM is done offline. The millisecond-level time overhead is negligible considering that the video segment request and head movement prediction are only performed every several seconds. Second, as prediction errors are always possible in head movement prediction, it is beneficial to stream a larger area than the predicted region based on the prediction accuracy in order to accommodate the errors. Alternatively, image-based rendering can be exploited to compensate the missing tiles. Since these strategies sacrifice either bandwidth efficiency or local computation efficiency, a careful tradeoff is necessary before the real-world deployment.

8 CONCLUSION

In this paper, we take an important step in exploring head movement prediction for 360-degree videos by leveraging the unique panoramic saliency. Motivated by the issues of central bias and multi-object confusion in traditional saliency models, we present PanoSalNet to detect panoramic saliency using the new self-built dataset. Then the proposed head movement prediction framework is trained based on the combination of panoramic saliency and head orientation history. With the accurate prediction at 0.5-1.0 second, it can potentially improve the performance of VR systems, e.g., providing a smooth playback by prefetching predicted views and expediting the rendering of users' viewport from the equirectangular format by caching previous computations.

We would like to emphasize that this research represents a significant attempt to address traditional vision problems on a new media content – 360-degree video. This work not only enriches computer vision research, but also boosts the performance of multimedia systems. We believe the success of this research can enable a suite of future works studying other interdisciplinary problems involving computer vision and multimedia systems.

9 ACKNOWLEDGEMENT

This work is supported by Intel AI DevCloud Usage for Research.

REFERENCES

[1] Ana De Abreu, Cagri Ozcinar, and Aljosa Smolic. 2017. Look Around You: Saliency Maps for Omnidirectional Images in VR Applications. In *IEEE International Conference on Quality of Multimedia Experience (QoMEX)*.

[2] A Deniz Aladagli, Erhan Ekmekcioglu, Dmitri Jarnikov, and Ahmet Kondoz. 2017. Predicting Head Trajectories in 360°Virtual Reality Videos. In *IEEE International Conference on 3D Immersion (IC3D)*.

[3] Marc Assens, Xavier Giro-i-Nieto, Kevin McGuinness, and Noel E. O'Connor. 2017. SaltiNet: Scan-path Prediction on 360 Degree Images using Saliency Volumes. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*.

[4] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Fredo Durand. 2018. What Do Different Evaluation Metrics Tell Us About Saliency Models? *IEEE Trans. Pattern Anal. Mach. Intell.* PP (March 2018), 1–1.

[5] Xavier Corbillon, Gwendal Simon, Alisa Devlic, and Jacob Chakareski. 2017. Viewport-adaptive Navigable 360-degree Video Delivery. In *IEEE International Conference on Communications (ICC)*.

[6] Xavier Corbillon, Francesca De Simone, and Gwendal Simon. 2017. 360-Degree Video Head Movement Dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference*.

[7] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. 2016. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*.

[8] International Data Corporation. https://goo.gl/DHb26g. Worldwide Quarterly Augmented and Virtual Reality Headset Tracker. (https://goo.gl/DHb26g).

[9] Fanyi Duanmu, Eymen Kurdoglu, S Amir Hosseini, Yong Liu, and Yao Wang. 2017. Prioritized Buffer Control in Two-tier 360 Video Streaming. In *ACM Workshop on Virtual Reality and Augmented Reality Network*.

[10] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2017. Fixation Prediction for 360 Video Streaming in Head-Mounted Virtual Reality. In *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*.

[11] Yuming Fang, Weisi Lin, Zhenzhong Chen, Chia-Ming Tsai, and Chia-Wen Lin. 2012. Video Saliency Detection in the Compressed Domain. In *ACM International Conference on Multimedia (MM)*.

[12] Yu Fang, Ryoichi Nakashima, Kazumichi Matsumiya, Ichiro Kuriki, and Satoshi Shioiri. 2015. Eye-head coordination for visual cognitive processing. In *PLoS ONE 10(3): e0121035. https://doi.org/10.1371/journal.pone.0121035.*

[13] Wenzhong Guo, Xiaolong Sun, and Yuzhen Niu. 2015. Evaluation of visual saliency analysis algorithms in noisy images. *IET Computer Vision* 9(2) (2015), 290–299.

[14] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. 2015. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *ICCV*.

[15] Tilke Judd, Krista Ehinger, FrÃĺdo Durand, and Antonio Torralba. 2009. Learning to predict where humans look. In *International Conference on Computer Vision (ICCV)*.

[16] Keras. https://keras.io. Keras: The Python Deep Learning library. (https://keras.io).

[17] Wolf Kienzle, Bernhard Scholkopf, Felix Wichmann, and Matthias Franz. 2007. How to Find Interesting Locations in Video: A Spatiotemporal Interest Point Detector Learned from Human Eye Movements. In *Hamprecht F.A., Schnorr C., Jahne B. (eds) Pattern Recognition. DAGM 2007. Lecture Notes in Computer Science.*

[18] J. B. Kuipers. 1999. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality.* Princeton University Press, Princeton, USA.

[19] Matthias Kummerer, Lucas Theis, and Matthias Bethge. 2015. Deep Gaze I: Boosting Saliency Prediction with Feature Maps Trained on ImageNet. In *International Conference on Learning Representations (ICLR)*.

[20] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient transfer learning. In *5th International Conference on Learning Representations (ICLR)*.

[21] Rafael Monroy, Sebastian Lutz, Tejo Chalasani, and Aljosa Smolic. 2017. SalNet360: Saliency Maps for omni-directional images with CNN. In *arXiv preprint arXiv:1709.06505v1*.

[22] Afshin Taghavi Nasrabadi, Anahita Mahzari, Joseph D. Beshay, and Ravi Prakash. 2017. Adaptive 360-Degree Video Streaming using Scalable Video Coding. In *ACM International Conference on Multimedia (MM)*.

[23] Tam V Nguyen, Mengdi Xu, Guangyu Gao, Mohan Kankanhalli, Qi Tian, and Shuicheng Yan. 2013. Static Saliency vs. Dynamic Saliency: A Comparative Study. In *ACM International Conference on Multimedia (MM)*.

[24] Yuzhen Niu, Lingling Ke, and Wenzhong Guo. 2016. Evaluation of visual saliency analysis algorithms in noisy images. *Machine Vision and Applications* 27(6) (2016), 915–927.

[25] Yuzhen Niu, Lening Lin, Yuzhong Chen, and Lingling Ke. [n. d.]. Machine learning-based framework for saliency detection in distorted images. *Multimedia Tools Application* ([n. d.]).

[26] Yuzhen Niu, Wenqi Lin, and Xiao Ke. 2018. CF-based optimisation for saliency detection. *IET Computer Vision* 12(4) (2018), 365–376.

[27] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. 2016. Shallow and Deep Convolutional Networks for Saliency Prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[28] Feng Qian, Lusheng Ji, Bo Han, and Vijay Gopalakrishnan. 2016. Optimizing 360 video delivery over cellular networks. In *ACM Workshop on All Things Cellular: Operations, Applications and Challenges*.

[29] Hasim Sak, Andrew Senior, and Francoise Beaufays. 2014. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. In *15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[30] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *arXiv preprint arXiv:1409.1556, 2014.*

[31] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. 2018. Saliency in VR: How do People Explore Virtual Environments? *IEEE Trans. Vis. Comput. Graphics* 24 (April 2018), 1633–1642.

[32] Benjamin W Tatler. 2007. The Central Fixation Bias in Scene Viewing: Selecting an Optimal Viewing Position Independently of Motor Biases and Image Feature Distributions. *Journal of Vision* 7 (Nov. 2007), 1–17.

[33] Technavio. https://goo.gl/zJCdnO. Global 360-Degree Camera Market 2016-2020. (https://goo.gl/zJCdnO).

[34] Evgeniy Upenik and Touradj Ebrahimi. 2017. A Simple Method to Obtain Visual Attention Data in Head Mounted Virtual Reality. In *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*.

[35] JMP Van Waveren. 2016. The asynchronous time warp for virtual reality on consumer hardware. In *ACM Conference on Virtual Reality Software and Technology (VRST)*.

[36] Shengke Wang, Shan Wu, Lianghua Duan, Changyin Yu, Yujuan Sun, and Junyu Dong. 2016. Person Re-Identification with Deep Features and Transfer Learning. In *arXiv:1611.05244*.

[37] Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang. 2017. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*.

[38] Mengbai Xiao, Chao Zhou, Yao Liu, and Songqing Chen. 2017. OpTile: Toward Optimal Tiling in 360-degree Video Streaming. In *ACM International Conference on Multimedia (MM)*.

[39] Lan Xie, Zhimin Xu, Yixuan Ban, Xinggong Zhang, and Zongming Guo. 2017. 360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming. In *ACM International Conference on Multimedia (MM)*.

[40] Yanhao Zhang, Lei Qin, Qingming Huang, Kuiyuan Yang, Jun Zhang, and Hongxun Yao. 2016. From Seed Discovery to Deep Reconstruction: Predicting Saliency in Crowd via Deep Networks. In *ACM International Conference on Multimedia (MM)*.