Gaze Prediction in Dynamic 360° Immersive Videos

Yanyu Xu, Yanbing Dong, Junru Wu, Zhengzhong Sun, Zhiru Shi, Jingyi Yu, and Shenghua Gao*
ShanghaiTech University
{xuyy2, dongyb, wujr1, sunzhy, shizhr, yujingyi, gaoshh}@shanghaitech.edu.cn

Abstract

This paper explores gaze prediction in dynamic 360° immersive videos, i.e.*, based on the history scan path and VR contents, we predict where a viewer will look at an upcoming time. To tackle this problem, we first present the large-scale eye-tracking in dynamic VR scene dataset. Our dataset contains 208 360° videos captured in dynamic scenes, and each video is viewed by at least 31 subjects. Our analysis shows that gaze prediction depends on its history scan path and image contents. In terms of the image contents, those salient objects easily attract viewers' attention. On the one hand, the saliency is related to both appearance and motion of the objects. Considering that the saliency measured at different scales is different, we propose to compute saliency maps at different spatial scales: the sub-image patch centered at current gaze point, the sub-image corresponding to the Field of View (FoV), and the panorama image. Then we feed both the saliency maps and the corresponding images into a Convolutional Neural Network (CNN) for feature extraction. Meanwhile, we also use a Long-Short-Term-Memory (LSTM) to encode the history scan path. Then we combine the CNN features and LSTM features for gaze displacement prediction between gaze point at a current time and gaze point at an upcoming time. Extensive experiments validate the effectiveness of our method for gaze prediction in dynamic VR scenes.*

1. Introduction

There is an emerging interest in viewing 360° VR contents in head-mounted displays (HMD) in place of a rectangular one on the screen. Commodity omnidirectional cameras such as Google Jump, Nokia OZO, Facebook 360, *etc.* are readily available to generate high quality 360° video contents and provide viewers with an immersive viewing experience. Social media platforms including YouTube and Facebook also support viewing 360° videos with or without HMDs. Same as 2D images and videos, the most intriguing problem in 360° videos, from both the commercial and technology perspectives, is to determine where a user would look at in the video. A successful solution will greatly benefit VR content creation.

The link between our task and 360° video setup. i) In gaze prediction on standard videos, users PASSIVELY watch the videos. In 360° immersive videos, users can ACTIVELY rotate the heads and body and decide where he looks. ii) Previous saliency detection in 360° videos watches STILL scene, so they can directly collect the eye fixation of all participants at the same scene for generating ground-truth. Our scene is DYNAMIC, for each frame, where a participant looks depends on its starting point and his decision on movement direction. So it is extremely difficult to annotate ground-truth for saliency detection. iii) our task is doable because we can collect the ground-truth, and this task is useful for many applications (as listed below).

This paper explores gaze prediction in 360° videos, aiming to study the user gaze behavior in 360° immersive videos. It shows the necessity and challenges for gaze prediction in this dynamic 360° immersive videos because of its importance for user behavior analysis while watching 360° VR videos and benefiting the data compression in VR data transmission [41]. Further, VR gaze prediction can also benefit applications beyond 360 videos: once we predict the viewing region of each participant viewer in upcoming frames, we can further improve user-computer interactions by tailoring the interactions for this specific viewer. In VR games, it is essential to effectively design different difficulty levels of the game for different players. For example, if we can reliably detect gaze in the current frame and predict its motion in future frames, we can change the degree of difficulty of the game on the fly: positioning rewards closer to the gaze region to make it easier or farther away to make it more difficult.

Despite tremendous efforts and achievements on saliency detection and gaze tracking, there is only a handful of work that focuses on studying how users watch a 360° video, largely due to difficulties in tracking gaze beneath the HMD: the eyes are covered by the HMD and traditional camera-based tracking is not directly applicable. The diffi-

*Corresponding author

Datasets	Scene	Videos	Video clip duration	Frames/Images	Viewers	Ground-truth annotation	HMD	Outputs
Sitzmann *et al.* [36]	Static	-	-	20	86	Eye tracking and Head movement in VR	Oculus DK2	Fixation points and Head position
Rai *et al.* [34]	Static	-	-	98	40-42	Eye tracking and Head movement in VR	Oculus DK2	Fixation points and Head position
Yu *et al.* [42]	Dynamic	10	10sec	2,500	10	Head movement in VR	Oculus DK2	Head position
Lo *et al.* [30]	Dynamic	10	60sec	15,000	50	Head movement in VR	Oculus DK2	Head position
Corbillon *et al.* [5]	Dynamic	7	70sec	16,450	59	Head movement in VR	Razer OSVR HDK2 HMD	Head position
360° Sports [19]	Dynamic	342	NA	180,000	5	Annotate salient object in panorama	Without using HMD	Manually labeled bounding box
Ours	Dynamic	208	20sec - 70sec	210,000	25	Eye tracking in VR	HTC VIVE	Fixation points and Head position

Table 1. The basic properties of the existing 360° video datasets.

Figure 1. The examples of our Dataset

culties in gaze tracking in VR setting consequently restricts gaze prediction in 360° videos. In this work, we employ an emerging in-helmet camera system, a *'7invensun a-Glass'* eye tracker, that is able to capture eye locations for conducting gaze tracking when a user views a specific frame in 360° videos. Then we embed it into an HTC VIVE headset. With this device, we create a large-scale VR gaze tracking dataset by collecting the eye fixation of viewers with a gaze tracker deployed in an HMD when they watch 360° videos in a real immersive VR setting (the users also wear the earphones when they watching VR videos). Our dataset consists of 208 360° video clips viewed by 30+ human subjects, and the length of the videos range from 20 to 60 seconds. Examples in Fig. 1 show some examples on different clips. With this dataset, we conduct the gaze prediction in VR videos.

Next, with this dataset collected with an HTC VIVE headset and *7invensun a-Glass* eye tracker, we conduct the gaze prediction in dynamic VR videos. Specifically, we present a deep learning based computational model towards robust VR gaze prediction. Recall that watching 360° videos is different from watching perspective 2D videos: in the former, a viewer actively chooses to the direction watch (e.g., by turning his or her head) whereas in the latter they can only view the video from a fixed pose. In other words, a viewer will have a much higher degree of freedom when watching 360° videos. Specifically, we leverage an LSTM module to estimate the viewer's behavior (watching pattern) under the fixed FoV. At the same time, we find that a viewer is more likely to be attracted by salient objects characterized by appearance and motion, thus we also take into the saliency into our consideration, specifically, we consider the saliency at different spatial scales in terms of video contents in an area centered at current gaze point, the video contents in the current FOV, and the video contents in the whole 360° scene. Then we feed the images as well as their saliency maps at different scales into a CNN. Then we combine the CNN features with LSTM features to predict the gaze displacement from current moment to next moment.

The contributions of our paper are three-fold: i) to our knowledge, it is the first work that specifically tackles the gaze prediction task in dynamic 360° video; ii) we construct the first large-scale eye tracking database on 360° videos with a gaze tracker deployed in an HMD in a real immersive VR setting; iii) we employ a saliency-driven solution for gaze prediction, and extensive experiments validate the effectiveness of our method.

2. Related Work

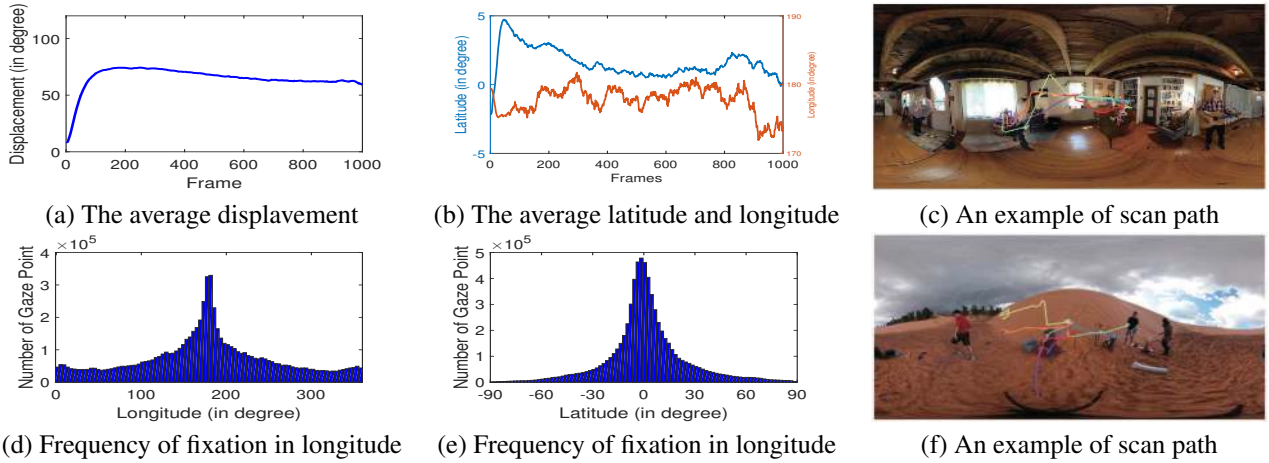
2.1. Saliency Detection

Tremendous efforts on saliency detection have been focused on predicting saliency map. In [1] Borji *et al.* provide a comprehensive study on existing saliency detection schemes. Most of the models are based on the bottom-up [10] [2] [14] [23], top-down [25] [13] [8] [22] [29], or hybrid approaches to detect salient regions on images. Recently, advances in deep learning have produced more accurate models [20] [33] [28] [37] [24]. Some work also attempts to use low-level appearance and motion cues as inputs or extend deep learning approaches to more complex scenarios such as stereo images [9] or videos [4] [7] [11] [31] [35] [32] [16] [12] [6] [18].

Though lots of work has been done for study saliency in image and video, saliency in VR is still in its primitive stage. Recently, [36] [34] also propose to study the VR saliency in static 360° images. However, the VR scenes are usually dynamic. Further, [19] propose to extract salient objects in VR videos, but the salient objects are manually annotated with panorama rather than obtained with gaze tracking in immersive VR. In this work, we leverage an a-Glass eye tracker for gaze tracking, we can capture the eye movements of viewers while they are experiencing in immersive VR.

2.2. Gaze Prediction on Egocentric Videos

In egocentric videos, camera wearer is usually an action doer, moving his/her head and interacting (touching, moving, etc.) with the objects. Gaze prediction under this setting is usually based on camera's rotation velocity/direction



(a) The average displacement (b) The average latitude and longitude (c) An example of scan path

(d) Frequency of fixation in longitude (e) Frequency of fixation in longitude (f) An example of scan path

Figure 2. Dataset Analysis: (a) and (b) shows the average intersection angle of gaze directions, the difference in latitude and longitude between two viewers over time when they are watching the same videos; (c) and (f) show some examples of viewers' scan path; (d) and (e) show the distribution of gaze points in latitude and longitude. (Best viewed in color)

of movement [39] and hand location [26]. In [44], a generator is used to generate future frames for gaze prediction. The scenes in [44] are indoor scene and relatively easy (e.g.,cutting food in the kitchen). In contrast, the observer of 360° videos could not interact with the objects in videos, so the hands information is not recorded. Further, the camera's motion information is not provided for use. So [26], [39] cannot be applied. Since our videos contain various scenes, including sports games, movies, music shows, etc. Compared with [44], the contents here are more diverse and change faster and significantly. Currently, video prediction itself is still an extremely challenging task in computer vision. The videos generated with [44] under our setting are very blurry and leads very poor results. It still needs future efforts to extend [44] for our task.

2.3. 360° Video Datasets

Some 360° video datasets have been created for viewer behavior analysis in VR, and based on the ground truth annotated in these datasets, they can be categorized into head movements analysis based datasets and gaze tracking based datasets. Head movements datasets only record the movements of heads, but even the heads are still, the viewers' eyeballs are still moving, this is, viewers still actively search the environment in their Field of View (FoV). So datasets in this category cannot provide the detailed eye movement information, and the datasets in this category are usually used for data compression for VR videos. In contrast, eye tracking based VR datasets provide the gaze points (eye fixation) at a different time. In [36] [34], eye tracker based gaze tracking has been introduced into the static VR scenes to study the viewers' behavior. However, given enough time, all viewers will explore the same scene even their moving trajectories are different, so we can integrate their eye fixation and get the saliency map. But for the dynamic VR scenes, viewers may be attracted by different moving objects if they are in immersive VR environment, different viewers may look at different scenes if their trajectories are different. Therefore it is more challenging than static scenes. To facilitate the ground truth annotation, in [19], Hu *et al.* build a large 360° sports videos dataset by asking viewers manually annotate the salient object with panorama rather than in HMD screen. However, the HMD based immersive VR experiencing is still different from the panorama. So we propose to use an aGlass eye tracker to record the gaze points when they are experiencing dynamic immersive VR and build the first gaze tracking dataset for dynamic immersive VR scenes. We summarize all these datasets in Table 1.

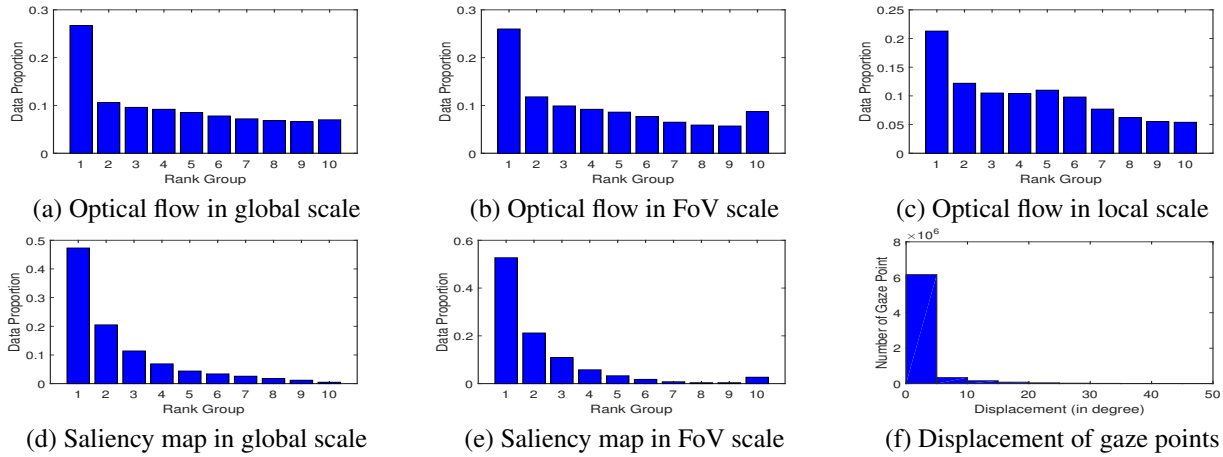
3. Dataset

In this section, we introduce a large-scale gaze tracking dataset for dynamic immersive 360° videos for VR viewer behavior analysis*.

3.1. Data Collection Protocol

Our dataset consists of 208 high definition dynamic 360° videos collected from Youtube, each with at least 4k resolution (3840 pixels in width) and 25 frames per second. The duration of each video ranges from 20 to 60 seconds. The videos in our dataset exhibit a large diversity in terms of contents, which include indoor scene, outdoor activities, music shows, sports games, documentation, short movies, *etc.* Further, some videos are captured from a fixed camera view and some are shotted with a moving camera that would probably introduce more variance in eye fixation

*https://github.com/xuyanyu-shh/VR-EyeTracking



(a) Optical flow in global scale (b) Optical flow in FoV scale (c) Optical flow in local scale

(d) Saliency map in global scale (e) Saliency map in FoV scale (f) Displacement of gaze points

Figure 3. Dataset analysis: (a) (b) and (c) show the coincidence of gaze points with the largest magnitude of optical flow with different scales; (d) and (e) show the coincidence of gaze points with saliency at different scales; (f) show the displacement between the gaze points at neighboring frames. (Best viewed in color)

across different users. Fig 1 shows example frames of 360° videos in our dataset.

We use an HTC VIVE as our HMD to play the 360° video clips, a '*7invensun a-Glass*' eye tracker is mounted within the HMD to capture the gaze of the viewer. 45 participants (25 males and 20 females) aging from 20 to 24 is recruited to take part in the experiment. All participants were reported normal or corrected-to-normal vision in the HMD setting and were instructed to freely explore in the video scene.

We divide all 208 video clips into 6 groups each containing around 35 video clips, participants only watch one group of the videos each time. Video scenes were played in a randomized order and the starting point is fixed (0° in latitude and 180° in longitude). To alleviate fatigue when viewing 360° videos, we enforce a short break (20sec) between two video clips and a long break (3 min) in the middle of a video group. The eye tracker is re-calibrated after each group when the participants take off the HMD. The total time for a participant to watch the videos in each group is approximately 30 minutes including calibration and break. The total time for collecting this dataset is about 100 hours. Even though some participants didn't watch all the videos in all groups, in the end, each video is watched by at least 31 participants. During the experiment, The Unity game engine was used to display all scenes and record the viewer's heading and gaze direction, we then intersect it with a 3D-Sphere and project it back to a panorama to acquire the corresponding heading and gaze coordinate on the video frames.

3.2. Dataset Analysis

3.2.1 Consistency of Eye Fixation

We calculate the inter-subject the difference of gaze trajectory among different viewers across all videos to measure the consistency of different viewers when they are experiencing immersive VR videos. Specifically, we enumerate all viewer pairs over each video, then calculate the average intersection angle of gaze directions, the difference in latitude and longitude between two viewers over time when they are watching the same videos. Then we average these measurements over all the videos. The results are shown in Fig. 2 (a) (b). We can see there exists heterogeneity of viewing trajectory of different viewers in dynamic 360° immersive videos, because of the change of video contents over time. Different viewers may be attracted by different contents, and their history gaze path also affects its future eye fixation.

3.2.2 Distribution of gaze points in latitude and longitude

To explore the gaze pattern of participants in VR environment, we plot the frequency of projected gaze coordinates in latitude and longitude respectively as shown in Fig. 2 (d) and (e). The plot shows an "equator bias" which agrees with the discovery in [36]. Our study also shows that the scatter of gaze points along longitude is more severe than that along latitude. The possible reason is that viewers usually tend to look left and right more frequently than up and down. We also compute the average changing time of direction along longitude for all viewers, which is 2.3. The small changing time indicates that viewers tend to explore the scene with the consistent direction along longitude.

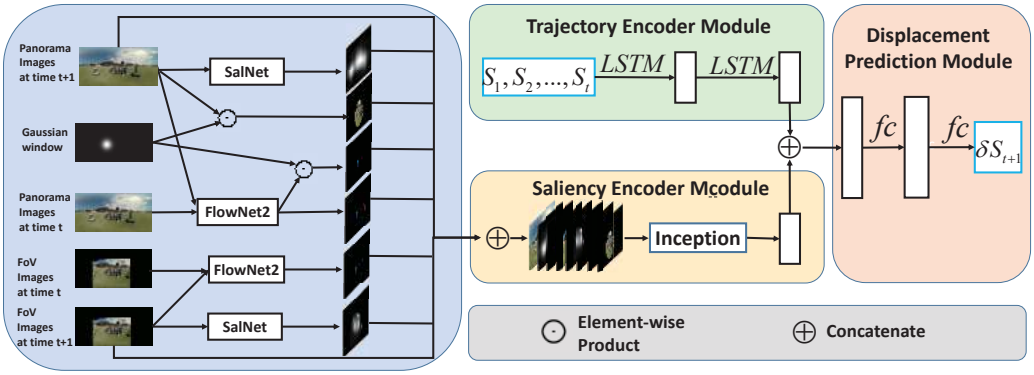


Figure 4. The architecture of our proposed deep neural network.

3.2.3 The relationship between gaze points and salient regions

Saliency detection assumes those more salient regions attract viewers attention. So we use the SalNet to calculate the spatial saliency map for each panorama image. Then we rank the saliency of all pixels in descending order. Based on the highest/lowest saliency value in each frame, we evenly divide these pixels into 10 bins. Based on the bin that the gaze point associated with the frame falls into, we get a frequency histogram of the gaze points fallen into different bins for all videos. The results are shown in Fig. 3 (d). We can see that gaze points usually coincide with salient points. Results based on FOV sub-image also show a similar phenomenon in Fig. 3 (e).

People sometimes are attracted by those moving objects, so temporal saliency is also an important factor for gaze tracking. Thus we calculate the optical flow with FlowNet2.0 [21], and the input of FlowNet2.0 is the panorama images of two consecutive frames. We also use the same way previously used to study the relationship between gaze points and motion. Results in Fig. 3 (a) show that gaze points usually coincide with pixels with large motion. Results based on FOV and local optical flow also show a similar phenomenon in Fig. 3 (b) and (c).

3.2.4 The distribution of angles corresponding to gaze points of neighboring frames

We also show the distribution of angles between gaze points of neighboring frames. The results are shown in Fig. 3 (f). We can see that usually the displacement between two temporally neighboring gaze points is very small. In other words, the gaze point of next frames falls into the neighbors of current gaze point.

4. Method

4.1. Problem Formulation

We formulate the gaze prediction in VR problem as follows: Given a sequence of 360° VR video frames $\mathbf{V}_{1:t} = \{v_1, v_2, \ldots, v_t\}$ where v_t corresponds to the t^{th} frame, and the gaze points of the p^{th} user corresponding to this video ($\mathbf{L}^p_{1:t} = \{l_1, l_2, \ldots, l_t\}$ where $l_t = (x_t, y_t)$, x_t and y_t is the latitude and longitude of the gaze intersection on a 3D-Sphere where $x_t \in [0, 360]$, $y_t \in [-90, 90]$), then gaze prediction aims to regress the future gaze coordinates corresponding to the future T frames:$\mathbf{L}_i$ where $i = t+1, ..., t+T$. It is worth noting that to simplify the problem, following the pioneer work in gaze prediction in 2D videos [44], currently we only sample one gaze point for each frame.

The gaze pattern in future frames is related to multiple factors. On the one hand, gaze points are largely correlated with spatial saliency which can be inferred from image contents, and temporal saliency which can be inferred from the optical flow between neighboring frames, as shown in Section 3.2 (Fig. 3). On the other hand, the users' history gaze path is also a key factor in predicting his/her future gaze point because different users have different habits in exploring a scene. For example, some users would look up and down frequently, and some users seldom look up or down. Another reason for leveraging users' history gaze path for gaze prediction is that users tend to explore the whole scene first by walking along the same longitude changing direction, *i.e.* , walking from left to right or from right to left consistently rather than frequently changing the walking direction, as shown in Section 3.2 (Fig. 2). The relationship between the gaze prediction and its history gaze path also motivates us to sequentially predict the gaze point for each future frame.

Based on above analysis, we formulate gaze prediction as a task of learning a nonlinear mapping function F which maps the history gaze path and image contents to the coordinates. Inspired by the recent success of residual learning

or displacement prediction in image classification [17], face alignment [43], and pose estimation [3], we propose to predict the displacement of gaze coordinates between the upcoming frame and current frame: $l_{t+1} - l_t$. Mathematically, we formulate the objective of gaze tracking as follows:

$$F^* = \arg\min_F \sum_{t=obs}^{obs+T-1} \|l_{t+1} - (l_t + F(\mathbf{V}_{t:t+1}, \mathbf{L}_{1:t}))\|^2 \quad (1)$$

where obs is the number of observed frames. It is worth noting that here we only consider the current frame and next frame and the history gaze path as input when we predict the gaze point in next frame. Two neighboring frames characterize the motion information (optical flow), and the next frame provides contents for saliency characterization. Then we use a deep neural network to model F (as shown in Fig. 4. Specifically, our network consists of a Trajectory Encoder module and a Saliency Encoder module, and a Displacement Prediction module. Next, we will detail these models sequentially.

4.2. Trajectory Encoder Module

Trajectory encoder module is designed to encode the history gaze path of a user. As aforementioned, viewers tend to explore the scene with the consistent direction along longitude. As for the gaze path direction along latitude, the gaze points usually are around the equator. In other words, the gaze path in history frames provides the clue for gaze prediction in future frames. In light of the good performance of LSTM networks for motion modeling [15], we also employ an LSTM network to encode the gaze pattern along the time.

For each video clip, we sequentially feed the gaze points (l_t^p) corresponding to history frames in this video sequence into a stacked LSTM, and denote the output of stacked LSTM f_{t+1}^p at $(t+1)^{th}$ frames,

$$f_{t+1}^p = h(l_1^p, l_2^p, ..., l_t^p) \quad (2)$$

Here the function $h(\cdot)$ represents the input-output function of stacked LSTM. In addition, the function h is stacked LSTMs with 2 LSTMs layers, both with 128 neurons.

4.3. Saliency Encoder Module

As aforementioned, gaze points usually coincide with spatial salient regions and objects with salient motions (large optical flows). In other words, saliency provides an important cue for gaze prediction in future frames. Thus we propose to incorporate the saliency prior regarding spatial saliency and temporal saliency which is characterized by optical flow features for gaze prediction. However, the saliency level of the same object at different spatial scales are different. So we propose to calculate the saliency with a multi-scale scheme, *i.e.* , i) local saliency: the saliency of local patch centered at current gaze point; ii) FOV saliency: the saliency of sub-image corresponding to current Field of View (FOV); and iii)Global saliency: the saliency of the global scene. In our implementation, we use the FlowNet2.0 [21] to extract the motion feature, and the input of FlowNet2.0 is the panorama images of two consecutive frames.

Local saliency. A viewer's gaze point at next frame depends on the coordinates of current gaze points. So we first generate a Gaussian window centered at current gaze point and use it to do the inner product with panorama image and optical flow map and use it as the spatial and temporal local saliency maps. It is worth noting that the local image patch is usually very small, and it usually doesn't contain a complete object. Rather than using saliency detection method to calculate the saliency, which is time-consuming, and usually cannot get very satisfactory saliency detection results because such a small image patch does not contain a complete object, our gaussian based saliency approximation demonstrates effectiveness and efficiency. Further, Gaussian based local saliency is also on par with that of classical saliency detection based solution.

FOV saliency. When a user experiencing immersive VR in HMD, his/her view is restricted by the FOV of the HMD. So it is necessary to take FOV saliency into consideration. Specifically, we calculate the saliency map corresponding to the sub-image in FOV with SalNet [33] which is a state-of-the-art saliency detection method. As for the motion saliency, we just set the optical flow of pixels outside of the FOV to 0.

Global saliency. Since the user can actively explore the VR scene, so it is possible he/she moves his/her head at next moment, and FOV will change consequently. So besides the local saliency and FOV saliency, global saliency is also necessary for understanding the scene. Similar to FOV saliency, we also feed the panorama image into SalNet for spatial saliency detection, and use the optical flow calculated with the panorama images as a temporal saliency estimation.

Previous work [27] has demonstrated the effectiveness of a coarse-to-fine strategy for saliency detection, *i.e.* , feeding an initial saliency result together with the original RGB image into a network for better saliency prediction. Motivated by this work, we propose to concatenate the RGB images, all spatial and temporal saliency maps, and feed them into an Inception-ResNet-V2 [38] to extract saliency features for gaze prediction. We denote $z(\cdot)$ represents the Inception V2 network, and denote S_{t+1}^p as all spatial and temporal saliency maps, and denote the saliency features as g_{t+1}^p, then g_{t+1}^p can be obtained as follows:

$$g_{t+1}^p = z(v_{t+1}, S_{t+1}) \quad (3)$$

where $z(\cdot)$ represents the subnet from input to the layer after

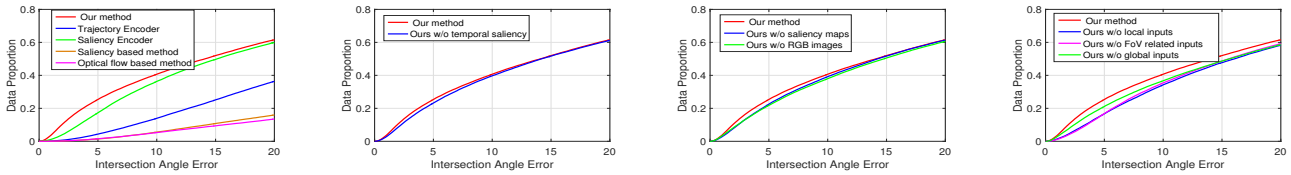


Figure 5. The effect of different components in our model. (Best viewed in color)

global pooling in Inception-ResNet-V2.

4.4. Displacement Prediction Module

The displacement prediction module takes the output of saliency encoder module and trajectory encoder module, use another two fully connected layer to estimate the displacement between the gaze point at time $t+1$ and gaze point at time t:

$$\delta l_{t+1}^p = r([f_{t+1}^p; g_{t+1}^p]). \quad (4)$$

, where $r(\cdot)$ represents two connected layers. The function r contains two fully connected layers with 1000, 2 neurons, respectively. Once we get the location displacement, we can compute the gaze coordinate l_{t+1}^p at time $t+1$: $l_{t+1}^p = l_t^p + \delta l_{t+1}^p$. We train the model by minimizing this loss across all the persons and all video clips in the training-set.

5. Experiment

5.1. Experimental Setup

Following the common setting in trajectory prediction for crowd [40], we downsample one frame from every five frames for model training and performance evaluation. In this way, the interval between two neighboring frames in our experiments corresponds to $\frac{5}{25}$ seconds, and such setting makes our gaze prediction task more challenging than that for the neighboring frames in original videos. In the following sections, the frames mentioned correspond to the sampled ones. Further, we propose to use the history gaze path in the first five frames to predict the gaze points in next five frames ($obs = 5$ and $T = 5$). We use the observation (the results of gaze tracking) in the first 1 second to predict the gaze points in the frames of upcoming 1 second.

We implement our solution on the TensorFlow framework. We train our network with the following hyper-parameters setting: mini-batch size (8), learning rate (0.1), momentum (0.9), weight decay (0.0005), and the number of epoch (5000). In our experiments, we randomly select 134 videos as training data, and use the remaining 74 videos as testing. Some participants are shared in training and testing, but the videos in training/testing have no overlap.

We propose to use the viewing angle between the predicted gaze point and its ground truth to measure the performance of gaze prediction. A smaller angle means the predicted gaze point agrees its prediction better. Since our goal aims at predicting a sequential gaze points in a video for each user, so we use the Mean Intersection Angle Error(MA) over all videos and all users to measure the performance. For the gaze point in the i^{th} frame ($i = 1, ..., T$) with ground truth (x_i^p, y_i^p) and prediction ($\hat{x}_i^p, \hat{y}_i^p$), the viewing angle between them can be represented as d_i (the way to compute d_i will be provided in supplementary material), then $MIAE$ can be calculated as follows $MIAE = \frac{1}{TP}\sum_{i,p} d_i$. Here P is the total number of users watching this video. We then average MAE over all videos. Following the work of face alignment [43], we also use cumulative distribution function (CDF) of all gaze points for performance evaluation. A higher CDF curve corresponds to a method with smaller MAE.

5.2. Performance Evaluation

Since there is no previous work on gaze prediction in dynamic VR scene. We design the following baselines: i) saliency based method: we use the location corresponding to the highest saliency in FOV as gaze prediction; ii) optical flow based method: we use the location corresponding to the highest magnitude in terms of optical flow as gaze prediction. iii) saliency encoder: we feed all the saliency maps and RGB images in different scales into saliency encoder for gaze point prediction; iv) trajectory encoder only: we only feed the history gaze path into trajectory encoder for gaze prediction; v) our model: we employ all the components for gaze point prediction.

The results of these baseline methods are listed in Fig. 5 (a), we can see that our method achieves the best of in terms of CDF. Further, image saliency method and optical flow based method do not consider the interaction between spatial and temporal saliency, which both attract users' visual attention. Both saliency encoder baseline and trajectory encoder baseline do not take all factors related to trajectory prediction, thus corresponds to poor performance. We also show some predicted gaze points in Fig. 6.

5.3. Evaluation of Different Components in Our Mothed

Evaluation of the necessity of temporal saliency In order to validate the effectiveness of temporal saliency, we train a network without optical flow to estimate the dis-

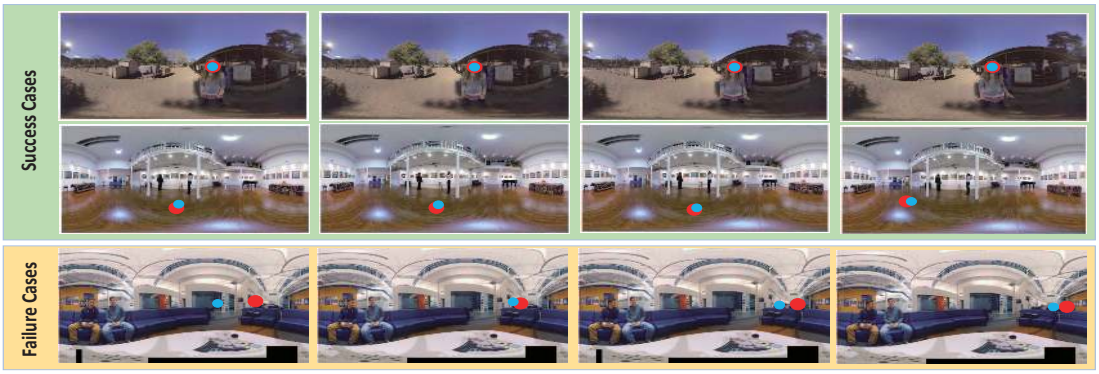


Figure 6. Qualitative results: ground truth (red, the bigger one), and predicted trajectories from our model (blue, the smaller). The first two rows show some successful cases and last row shows some failure cases.

placement of viewing angles. We compare the performance with/without temporal saliency in Fig. 5 (b). We can see that the network with temporal saliency performs better than the one without temporal saliency. This is because the network with temporal saliency takes the different importance of different moving objects into consideration. This agrees with our findings on our dataset that pixels with a higher magnitude of optical flow usually coincident with gaze points.

Evaluation of the necessity of RGB images and saliency maps in the input of saliency encoder We propose to remove the RGB image and saliency maps in the inputs of saliency encoder in our framework. Fig 5 (c) shows that after removing RGB images or saliency maps, the performance drops, this agrees with previous work [27] that the combination of initial saliency map and RGB image leads to better saliency features.

Evaluation of the necessity of multi-scale inputs To validate the effectiveness of multi-scale inputs, we train three networks by removing the inputs corresponds to local saliency, FOV related inputs, and inputs related to the global scene, respectively. The comparison is shown in Fig. 5 (d). We can see the network with multi-scale inputs achieves better performance, which validates the effectiveness of multi-scale inputs for extracting features.

5.4. Coordinates regression vs. displacement regression

In our gaze prediction model, we use a multi-layer perceptron to estimate the displacement δl_{t+1} between viewing angles at time $t+1$ and time t. Besides δl_{t+1} regression, we also train a model to directly estimating l_{t+1} from the same inputs. Specifically, the MAE of displacement based and directly coordinate based gaze prediction is 20.96 and 30.72, respectively. The excellent performance of displacement regression strategy validates the effectiveness of residual regression in gaze point prediction, which agrees with existing work for image classification and facial/body key points detection [43], and pose estimation [3].

5.5. Time costs.

Our model is implemented on an NVIDIA Tesla M40 GPU platform with an Intel(R) Xeon(R) CPU E5-2690 v4 2.60GHz CPU. We run our program 90 times and calculate the average running time for each image. The average running time of our model is 47 ms. If the model runs on CPU, the time cost is 466 ms.

6. Conclusion and Future Work

Our work attempts to understand how users experience a dynamic 360° immersive videos. We have built the first large-scale eye-tracking dataset for dynamic 360° immersive videos, and analyze the dataset in details. Our analysis shows that temporal and spatial saliency, as well as history gaze path, are three important factors for gaze prediction. Then we propose to use the dataset for gaze prediction with a deep learning framework by taking all these factors into consideration. Extensive experiments validate the effectiveness of our method. It is worth noting that there is still space to improve our method for gaze tracking. For example, currently, to simplify the problem, we only consider the motion in a short time interval for gaze prediction. Considering the motion in a longer time may further boost the performance. Sound is a very important factor. All videos in our datasets have sound information, and all participants wore earphone when watching 360° videos. The representation of sound in 360° videos is still an open task. We will take sound into consideration in our future work.

7. Acknowledgement

This project is supported by the NSFC (No. 61502304) and Program of Shanghai Subject Chief Scientist (A type) (No.15XD1502900).

References

[1] A. Borji, M. M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *Eprint Arxiv*, 16(7):3118, 2014.

[2] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? 2016.

[3] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016.

[4] S. Chaabouni, J. Benois-Pineau, O. Hadar, and C. B. Amar. Deep learning for saliency prediction in natural video. 2016.

[5] X. Corbillon, F. De Simone, and G. Simon. 360-degree video head movement dataset. pages 199–204, 2017.

[6] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. A deep multi-level network for saliency prediction. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3488–3493. IEEE, 2016.

[7] X. Cui, Q. Liu, and D. Metaxas. Temporal spectral residual:fast motion saliency detection. In *International Conference on Multimedia 2009, Vancouver, British Columbia, Canada, October*, pages 617–620, 2009.

[8] A. Gabadinho, N. S. Mller, M. Studer, J. D. Leeuw, and A. Zeileis. Analyzing and visualizing state sequences in r with traminer. *Journal of Statistical Software*, 40(04), 2011.

[9] A. Gibaldi, M. Vanegas, P. J. Bex, and G. Maiello. Evaluation of the tobii eyex eye tracking controller and matlab toolkit for research. *Behavior Research Methods*, 49(3):923–946, 2016.

[10] S. Goferman, L. Zelnikmanor, and A. Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 34(10):1915, 2012.

[11] C. Guo, Q. Ma, and L. Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *Ieee,conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[12] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE transactions on image processing*, 19(1):185–198, 2010.

[13] S. S. Hacisalihzade, L. W. Stark, and J. S. Allen. Visual perception and sequences of eye movement fixations: a stochastic modeling approach. *IEEE Transactions on Systems Man & Cybernetics*, 22(3):474–481, 1992.

[14] H. Hadizadeh and I. V. Bajić. Saliency-aware video compression. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 23(1):19, 2014.

[15] Y. D. B. Z. Hang Su, Jun Zhu. Forecast the plausible paths in crowd scenes. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2772–2778, 2017.

[16] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2007.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] S. Hossein Khatoonabadi, N. Vasconcelos, I. V. Bajic, and Y. Shan. How many bits does it take for a stimulus to be salient? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2015.

[19] H. N. Hu, Y. C. Lin, M. Y. Liu, H. T. Cheng, Y. J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360deg sports video. 2017.

[20] X. Huang, C. Shen, X. Boix, and Q. Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *IEEE International Conference on Computer Vision*, pages 262–270, 2015.

[21] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. 2016.

[22] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219, 1985.

[23] K. Koehler, F. Guo, S. Zhang, and M. P. Eckstein. What do saliency models predict? *Journal of Vision*, 14(3):14–14, 2014.

[24] S. S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, and R. V. Babu. Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5781–5790, 2016.

[25] X. Li. Learning to detect stereo saliency. (4):1–6, 2014.

[26] Y. Li, A. Fathi, and J. M. Rehg. Learning to predict gaze in egocentric video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3216–3223, 2013.

[27] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–686, 2016.

[28] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 362–370, 2015.

[29] R. Liu, J. Cao, Z. Lin, and S. Shan. Adaptive partial differential equation learning for visual saliency detection. In *Computer Vision and Pattern Recognition*, pages 3866–3873, 2014.

[30] W. C. Lo, C. L. Fan, J. Lee, C. Y. Huang, K. T. Chen, and C. H. Hsu. 360 video viewing dataset in head-mounted virtual reality. In *ACM on Multimedia Systems Conference*, pages 211–216, 2017.

[31] V. Mahadevan and N. Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 32(1):171–7, 2010.

[32] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, 3(1):5–24, 2011.

[33] J. Pan, E. Sayrol, X. Giroinieto, K. Mcguinness, and N. E. Oconnor. Shallow and deep convolutional networks for saliency prediction. pages 598–606, 2016.

[34] Y. Rai, J. Gutiérrez, and P. Le Callet. A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 205–210. ACM, 2017.

[35] H. J. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. *Journal of Vision*, 9(12):1–27, 2009.

[36] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, and G. Wetzstein. Saliency in vr: How do people explore virtual environments? 2016.

[37] K. Sss, K. Ayush, and R. V. Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing*, PP(99):1–1, 2015.

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[39] K. Yamada, Y. Sugano, T. Okabe, Y. Sato, A. Sugimoto, and K. Hiraki. Attention prediction in egocentric video using motion and visual saliency. In *Pacific-Rim Symposium on Image and Video Technology*, pages 277–288. Springer, 2011.

[40] S. Yi, H. Li, and X. Wang. Pedestrian behavior understanding and prediction with deep neural networks. In *European Conference on Computer Vision*, pages 263–279. Springer, 2016.

[41] M. Yu, H. Lakshman, and B. Girod. A framework to evaluate omnidirectional video coding schemes. In *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, pages 31–36. IEEE, 2015.

[42] M. Yu, H. Lakshman, and B. Girod. A framework to evaluate omnidirectional video coding schemes. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 31–36, 2015.

[43] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *European Conference on Computer Vision*, pages 1–16. Springer, 2014.

[44] M. Zhang, K. T. Ma, J. H. Lim, Q. Zhao, and J. Feng. Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4372–4381, 2017.