

Modelling Attention in Panoramic Video: A Reinforcement Learning Approach

Michael Shell, *Member, IEEE*, John Doe, *Fellow, OSA*, and Jane Doe, *Life Fellow, IEEE*

Abstract—Panoramic video provides immersive and interactive experience, by enabling human to control the field of view (FoV) with head movement (HM). Thus, HM plays a key role in modelling human attention on panoramic video. In this paper, we establish a database collecting subjects' HM positions on panoramic video sequences, and we find from our database that the HM data are highly consistent across subjects. We further find that deep reinforcement learning (DRL) can be applied in predicting HM positions, seen as actions of a DRL agent. Based on our findings, we propose a DRL based HM prediction (DHP) approach with offline and online versions, called offline-DHP and online-DHP, respectively. In offline-DHP, multiple DRL workflows are run to decide some potential HM positions on each panoramic video frame. Then, the heat map of the predicted HM positions, called the HM map, is generated as the output of offline-DHP. In online-DHP, the next HM position of one subject is estimated, given the observed HM positions of this subject till the current frames. Such online estimation is achieved by developing a DRL algorithm, based on the learned model of offline-DHP. Finally, experimental results validate that offline-DHP and online-DHP are effective in predicting HM positions of panoramic video in offline and online manners, respectively. Experimental results also show that the learned offline-DHP model is capable of improving the performance of online-DHP.

Index Terms—Computer Society, IEEE, IEEEtran, journal, LATEX, paper, template.

1 INTRODUCTION

DURING the past years, panoramic video [1] has been increasingly popular, due to its immersive and interactive experience. To achieve the immersive and interactive experience, human can control the field of view (FoV) via wearing head mounted displays (HMD), when watching panoramic video in a range of $360^\circ \times 180^\circ$. In other words, humans are able to freely move their heads within a sphere, to make their FoVs focus on the attractive content (see Figure 1 for an example). The content outside FoV cannot be seen by humans, i.e., without any attention. Consequently, head movement (HM) plays a key role in deploying human attention on panoramic video. HM prediction thus emerges as an increasingly important problem in modelling attention in panoramic video. Given the predicted HM, visual attention within FoV can be further modelled by the conventional saliency detection methods [2] of 2D video. The same as traditional 2D video, attention model can be extensively utilized in many areas of panoramic video, such as region-of-interest (ROI) compression [3], visual quality assessment [4], [5], rendering [6], synopsis [7], and automatic cinematography [8].

Unfortunately, few work has been proposed to model human attention on panoramic video, especially predicting the positions of HM. Benefiting from the most recent success of deep reinforcement learning (DRL) [9], this paper proposes a DRL based HM prediction (DHP) approach for modelling attention on panoramic video. In fact, HM prediction can be classified into two categories: offline and online manners. The offline HM prediction refers to modelling attention on panoramic video for multiple subjects, while the online prediction aims at predicting the next HM position of one subject upon the ground-truth of his/her HM positions at the current

and previous frames. In this paper, our DHP approach includes both online and offline HM prediction, named as offline-DHP and online-DHP, respectively.

To our best knowledge, there exists no offline work to predict HM positions of multiple subjects in viewing panoramic video. The closest work is saliency detection on 2D video [2]. The earliest approach for saliency detection was proposed by Itti *et al.* [10], in which the features of color, intensity and orientation are combined to generate the saliency map of an image. Later, Itti *et al.* [11] proposed to add two features in [10], motion and flicker contrast, for video saliency detection. Recently, several advanced approaches have been proposed for video saliency prediction. These advanced works include the earth mover's distance (EMD) approach [12] and the boolean map based saliency model (BMS) [13]. Most recently, deep learning has been successfully applied in saliency detection, such as SALICON [14] and Liu's approach [15]. Saliency detection in 2D video assumes that humans are able to view all content of each video frame. However, this assumption does not hold for panoramic video, as subjects can only see a limited range of FoV at a single sight, rather than the full panoramic range of $360^\circ \times 180^\circ$. In fact, different regions of panoramic video are accessible to subjects via changing the positions of HM [16]. In this paper, we find that different subjects are highly consistent on HM positions. Such finding is based on establishing and analyzing a new database, which consists of 58 subjects' HM data in viewing 76 panoramic video sequences. Then, we propose the offline-DHP approach to predict the consistent HM positions on panoramic video via generating the HM map for each single frame. The HM maps are in the form of sphere, and the positions in the HM maps are thus represented by the longitude and latitude in the geographic coordinate system (GDS) [17]. This paper visualizes the spherical HM maps by projecting them to the 2D plane. Figure 1 demonstrates an example of the ground-truth HM map for a panoramic video frame. Similar to the saliency maps of 2D video, the HM maps of panoramic video are obtained by convoluting HM

• *M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332.
E-mail: see <http://www.michaelshell.org/contact.html>*

• *J. Doe and J. Doe are with Anonymous University.*

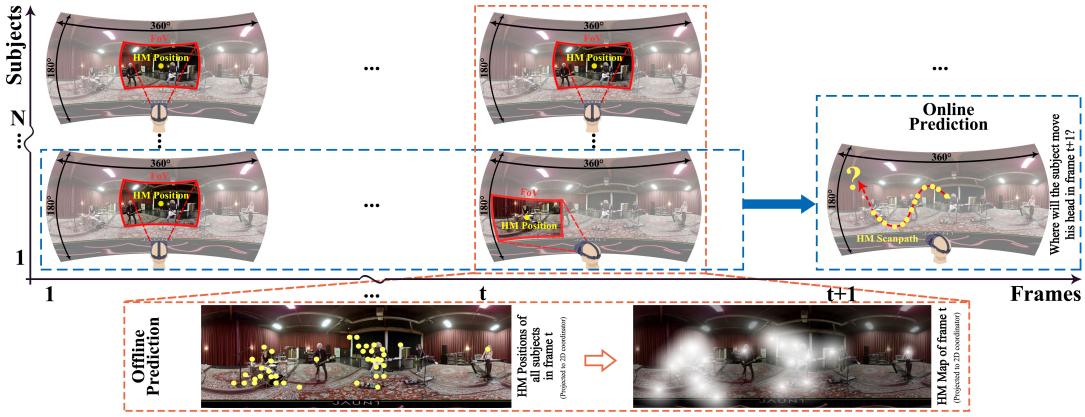


Fig. 1: Illustration for FoVs and HM positions across different subjects. The heat map of HM positions from all subjects is also shown, which is defined as the HM map.

positions with the 2D Gaussian filter¹.

Specifically, our offline-DHP approach yields the HM maps of panoramic video via predicting HM scan-paths of multiple *agents*, since subjects interactively control their HM positions along with some scan-paths according to video content. First, we find from our database that the HM scan-paths of different subjects are with high consistency. Meanwhile, subjects are normally initialized to view the center of front region in the beginning frames of panoramic video. Thereby, HM positions at the subsequent frames can be yielded on the basis of the predicted scan-paths. Additionally, we find from our database that the magnitudes and directions of HM scan-paths are with similarity across subjects. In light of these findings, our offline-DHP approach models both magnitudes and directions of HM scan-paths as the *actions* of multiple DRL *agents* and takes viewed panoramic content as the *observation* of *environment*. As such, the DRL model can be learned to predict HM positions. In training the DRL model, *reward* is designed to measure the difference of *actions* (i.e., HM scan-paths) between the DRL *agents* and subjects. Then, the *reward* is optimized to learn parameters in the DRL model. Given the learned model, the HM maps of panoramic video are generated upon the predicted HM positions, obtained from scan-paths of several *agents* in multiple DRL workflows.

For online HM prediction, the latest work of [18] proposed a deep 360 pilot, which automatically shifts viewing direction (equivalent to the HM position) in watching panoramic video. Specifically, the salient object is detected and tracked across panoramic video frames, via leveraging region-based convolutional neural network (RCNN) [19] and recurrent neural network (RNN). Given the detected salient object and previous viewing directions, the deep 360 pilot predicts to transit HM position by learning a regressor. Since the deep 360 pilot relies heavily on one salient object, it is only suitable for some specific scenes that include one salient object, e.g., the sport scenes in [18]. It is still challenging to predict HM positions online for generic panoramic video, which may include more than one salient object (e.g., the panoramic video of Figure 1). In this paper, we propose an online approach, namely online-DHP, to predict the HM positions on generic panoramic video. Different from [18], our online-DHP approach does not

need to detect the salient object by RCNN. Instead, it is based on attention-related content by leveraging the learned model of our offline-DHP approach. Then, a DRL algorithm is developed in our online-DHP approach to predict the HM positions in an online manner. Specifically, in the DRL algorithm, the *agent* predicts the *action* of HM scan-path in the next frame, according to ground-truth of the previous HM scan-path and *observation* of video content. Consequently, the HM positions at the incoming frames can be predicted for our online-DHP approach.

The main contributions of this paper are three-fold:

- We establish a new panoramic video database comprising HM positions of 58 subjects across 76 panoramic video sequences, with a thorough analysis on their HM data.
- We propose an offline-DHP approach to detect HM maps of panoramic video, which predicts the consistent HM positions of multiple subjects.
- We develop an online-DHP approach to predict the HM position of one subject at the next panoramic frame, upon video content and HM positions till the current frame.

2 RELATED WORK

2.1 Saliency detection

The only approach on predicting HM positions of panoramic videos is the most recent work of [8], in which Pano2Vid was proposed to yield FoV at each panoramic video frame. However, Pano2Vid mainly focuses on virtually generating a potential HM position at one frame, rather than modelling HM maps of multiple subjects at this frame. The closest work on predicting HM maps is saliency detection for 2D video, which is briefly reviewed in the following.

Saliency detection aims to predict visual attention of humans on 2D videos, by generating the saliency maps of video frames. The studies on visual saliency start from 1998, when Itti and Koch [10] found that the features of intensity, color and orientation in an image can be employed to detect its saliency map. Afterwards, they extended their work to video saliency detection [11], in which two dynamic features of motion and flicker contrast are combined with [10] to detect saliency in 2D videos. Both [10] and [11] can be seen as heuristic approaches to detect saliency, since they make use of the understanding of the HVS to develop the computational models. Recently, some advanced heuristic approaches, e.g., [12], [13],

1. The two dimensions of the Gaussian filter are longitude and latitude, respectively.

[20], [21], [22], [23], [24], [25], [26], have been proposed to detect saliency in 2D videos. Specifically, [20] proposed a novel feature called *surprise*, which measures how the visual change attracts human observers, based on the Kullback-Leibler (KL) divergence between spatio-temporal posterior and prior beliefs. Given the feature of *surprise*, a Bayesian framework was developed in [20] for video saliency detection. Some other Bayesian frameworks [21], [22] were also developed to detect video saliency. Besides, Lin *et al.* [12] quantified earth mover's distance (EMD) to measure the center-surround difference in spatio-temporal receptive field, generating saliency maps for 2D videos. Zhang *et al.* [13] explored the surround cue for saliency detection, by characterizing a set of binary images with randomly thresholds on color channels. Recently, [25] and [26] have investigated that some features (e.g., motion vector) in compressed domain are of high correlation with human attention, and they are thus explored in video saliency detection.

Benefiting from the most recent success of deep learning, deep neural networks (DNNs) [14], [15], [27], [28], [29], [30], [31] have also been developed to detect 2D video saliency, instead of exploring the HVS related features in heuristic saliency detection approaches. They can be seen as data-driven approaches. For static saliency detection, SALICON [19] fine-tuned the existing convolutional neural networks (CNN), with a new saliency related loss function. For dynamic saliency detection, [29] leveraged a deep convolutional 3D (C3D) network to learn the representations of human attention on 16 consecutive frames, and then a long short-term memory (LSTM) network connected with a mixture density network was learned to generate saliency maps in Gaussian mixture distribution. Similarly, Liu *et al.* [15] combined CNN and multi-stream LSTM to detect saliency in videos with multiple faces. Besides, other DNN structures have been developed to detect either static saliency [27], [28] or dynamic saliency [29], [30], [31].

Although saliency detection has been thoroughly studied in predicting eye movement on 2D videos, there is no work to predict HM positions on panoramic videos. Similar to saliency detection in 2D videos, this paper proposes to generate HM maps, which represent HM positions of multiple subjects. Hence, the HM maps model attention on panoramic videos. To obtain the HM maps of panoramic video, a DRL approach is developed for deciding the *actions* of HM from multiple *agents*. The *actions* are made upon the *environment* of panoramic video content, the features of which are automatically learned and then extracted by DNN. Thus, our approach takes advantage of both deep learning and reinforcement learning, driven by the HM data of our panoramic video database. It is worth mentioning that although few work applies DRL to predict human attention, attention model is widely used in the opposite direction, to improve the performance of reinforcement learning, e.g., [32], [33], [34], [35].

2.2 Virtual cinematography

Virtual cinematography of panoramic video was proposed in [8], [18], [36], [37], [38], which directs an imaginary camera to virtually capture natural FOV (NFOV). In general, virtual cinematography attempts to agree with HM positions of human at each panoramic video frame. The early work of [36] proposed cropping object-of-interest in panoramic video, such that NFOV can be generated for virtual cinematography. Later, in [37], the cropped object-of-interest is tracked across frames by a Kalman filter, for automatically controlling virtual camera in virtual cinematography of panoramic

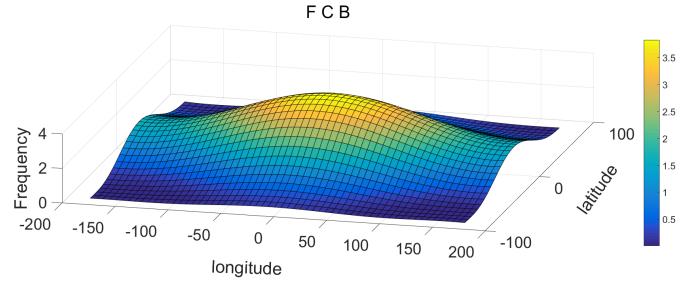


Fig. 2: The distribution of 58 subjects' HM positions calculated from all panoramic video frames of the PVS-HM database.

video. The approach of [37] can work on both compressed and uncompressed domains, as two methods were developed to detect object-of-interest in compressed and uncompressed domains, respectively. Both the works of [36], [37] were designed for the task of online virtual cinematography. They can be seen as heuristic approaches, which are not trained or even evaluated on the ground-truth HM data of human subjects.

Most recently, data-driven approaches have boosted the development of virtual cinematography for panoramic videos. Specifically, Pano2Vid [8] learns to generate NFOV at each panoramic frame. However, the learning mechanism of Pano2Vid is offline. In fact, NFOV can be estimated at each frame in an online manner, which uses observed HM positions of the previous frames to correct the estimation of NFOV at the current frame. To this end, online virtual cinematography [18], [38] has been studied in a data-driven way. Specifically, a state-of-the-art virtual cinematography approach, the deep 360 pilot, was proposed in [18], which is a deep learning based *agent* smoothly tracking object-of-interest for panoramic video. In other words, the *agent* transits the HM position across video frames to track the key object detected by RCNN, given the observed HM positions at previous frames. Consequently, NFOV can be generated online for automatically displaying object-of-interest in virtual cinematography of panoramic videos. In fact, object-of-interest tracking in panoramic video refers to continuously focusing and refocusing intended targets, respectively. Both focusing and re-focusing require a subject to catch up the object. Such a task is challenging in extreme-sport videos, as the object-of-interest may be moving fast. Therefore, Lin *et al.* [38] investigated two focus assistance techniques to help the subject track the key object in viewing panoramic video, in which the potential HM position attended to the object-of-interest needs to be determined and provided for the subject.

The above approaches of [8], [18], [36], [37], [38] all depend on the detector of object-of-interest. Thus, they can be only applied in some specific panoramic videos with salient objects, such as video conferencing or classroom scenes in [36], [37] and the sports videos in [8], [18], [38]. Different from these conventional approaches, our online-DHP approach is based on the learned model of our offline approach, which encodes HM related content rather than detecting object-of-interest. Consequently, our approach is object-free, thus more suitable for the generic panoramic videos.

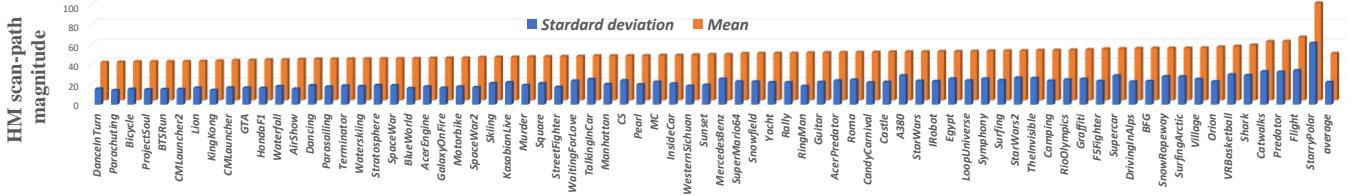


Fig. 3: Mean and standard deviation for the HM scan-path magnitude over different subjects, for 76 panoramic video sequences in the PVS-HM database. The last column also shows the mean and standard deviation, averaged over all sequences.

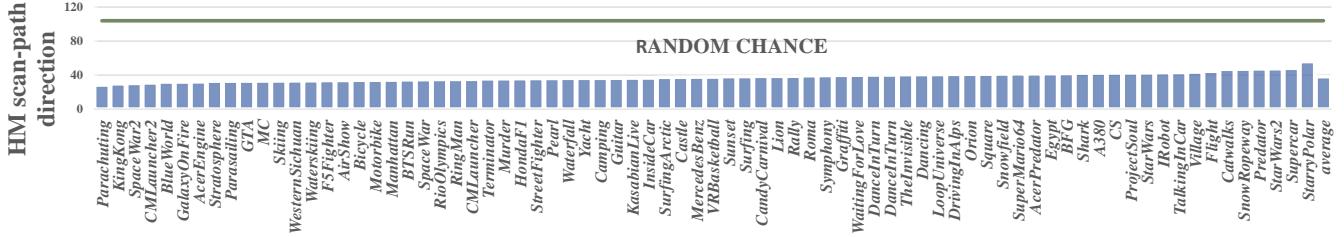


Fig. 4: Circular standard deviation for the direction of HM scan-paths, over 76 panoramic video sequences in the PVS-HM database. The average results are also shown in the last column.

3 DATABASE ESTABLISHMENT AND ANALYSIS

3.1 Database establishment

In this section, we collect a new database including 76 panoramic video sequences with HM data of 58 subjects, called the PVS-HM database. Our PVS-HM database allows quantitative analysis of subjects' HM on panoramic video, and it can be also used for learning to predict where human looks at panoramic video. Our database is available in (XXX website XXX) for facilitating the future research. In the following, we present how we conducted the experiment to obtain the PVS-HM database.

First, we selected 76 panoramic video sequences from YouTube and VRCloud, with resolution ranging from 3K to 8K. As seen in Table 1, the content of these sequences is diverse, including computer animation (CA), driving, action sports, movie, video game, scenery, etc. Then, the duration of each sequence was cut to be 10 to 80 seconds (averagely 26.9 seconds), such that fatigue can be reduced in viewing panoramic video. To ensure video quality, all panoramic video sequences were compressed by H.265 [39] without any change at bit-rates. Note that the audio tracks were removed to avoid the impact of acoustic information on visual attention.

In our experiment, 58 subjects (41 males and 17 females, aging from 18 to 36) wore the HMD of HTC Vive to view all 76 panoramic video sequences at random display order. When watching panoramic video, the subjects seated on a swivel chair were allowed to turn around freely, such that all panoramic regions are accessible. To avoid eye fatigue and motion sickness, the subjects have a 5 minute rest after viewing each session of 19 sequences. With the support of the software development kit (SDK) of HTC Vive, we recorded the posture data of each subject when viewing panoramic video. Based on the recorded posture data, the HM data of all 58 subjects at each frame of the panoramic video sequences were obtained and stored for our PVS-HM database, in terms of longitude and latitude in the GDS.

3.2 Database analysis

In this section, we mine our PVS-HM database to analyze HM data of different subjects across panoramic video sequences. Specifically, we have the following six findings.

Finding 1: When watching panoramic video, different subjects are highly consistent in HM positions.

Analysis: In our PVS-HM database, we randomly divide all 58 subjects into two equal-size groups, *A* and *B*. For each frame of 76 sequences, the ground-truth HM maps of Groups *A* and *B* are generated by convolving with a 2D Gaussian filter over the collected HM data, along with longitude and latitude. They are denoted as \mathbf{H}_A and \mathbf{H}_B , respectively. For a panoramic frame, we quantify the correlation between \mathbf{H}_A and \mathbf{H}_B using linear correlation coefficient (CC) [40]. Table 1 lists the averaged CC (\pm standard deviations) between \mathbf{H}_A and \mathbf{H}_B over all frames, for each sequence. We further show in Table 1 the CC values between \mathbf{H}_A and the maps generated by Gaussian distribution, uniform distribution and FCB. It can be seen from this table that the CC values of Groups *A* and *B* are rather high, and that the averaged CC result of two groups is significantly higher than those of Gaussian distribution, uniform distribution and FCB. Thus, it is obvious that HM positions of different subjects are highly consistent. This completes the analysis of *Finding 1*.

Finding 2: The HM positions on panoramic video exist the front center bias (FCB).

Analysis: We investigate whether the FCB exists when people viewing panoramic video. The investigation is achieved by calculating the numbers of HM positions in different regions of video frames. First, the $360^\circ \times 180^\circ$ panoramic region of each frame is equally segmented to be $60^\circ \times 30^\circ$ grids. Then, the numbers of 58 subjects' HM positions are counted for each segmented grid. Consequently, the frequency of HM positions in different panoramic regions is obtained, and it is shown in Figure 2. Note that the frequency results of Figure 2 are obtained by averaging over all panoramic video frames of the PVS-HM database. We can see from this figure that

TABLE 1: CC between ground-truth HM maps of Groups *A* and *B*, for each panoramic video sequence

Category	Name	CC	Category	Name	CC	Category	Name	CC	Category	Name	CC
CA	AcerEngine	0.891±0.058	Others	A380	0.867±0.058	Driving	AirShow	0.883±0.033	Show	BTSRun	0.926±0.041
	AcerPredator	0.876±0.063		Camping	0.851±0.073		Bicycle	0.953±0.021		Catwalks	0.834±0.069
	BFG	0.732±0.101		CandyCarnival	0.830±0.088		DrivingInAlps	0.858±0.073		DanceInTurn	0.934±0.030
	CMLauncher	0.894±0.041		Lion	0.937±0.034		F5Fighter	0.572±0.109		Dancing	0.906±0.052
	CMLauncher2	0.924±0.045		MercedesBenz	0.685±0.137		HondaFI	0.944±0.020		Graffiti	0.906±0.037
	LoopUniverse	0.881±0.044		RingMan	0.923±0.032		InsideCar	0.903±0.036		Guitar	0.866±0.096
	Orion	0.810±0.082		Shark	0.812±0.116		Rally	0.925±0.025		KasabianLive	0.819±0.087
	Pearl	0.838±0.120		Square	0.842±0.076		Stratosphere	0.792±0.090		NotBeAloneTonight	0.695±0.104
	Roma	0.896±0.054		TalkingInCar	0.858±0.078		VRBasketball	0.739±0.098		Symphony	0.826±0.068
	Overall	0.896±0.054						0.830±0.075			

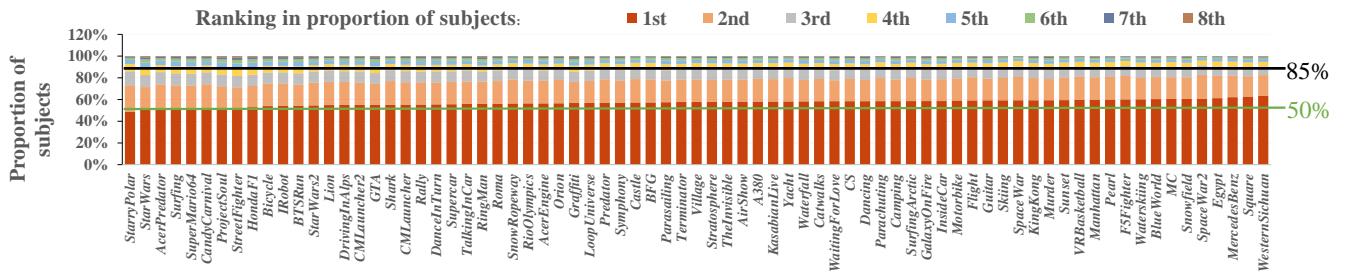


Fig. 5: Distribution of HM scan-path over 8 discrete directions, for all 76 sequences from the PVS-HM database. Note that the proportions of subjects, whose HM scan-path directions fall into each discrete direction, are ranked and then shown in this figure.

the HM positions of 58 subjects are more likely to be attracted by the regions of front center. This completes the analysis of *Finding 2*.

Finding 3: The magnitude of HM scan-paths is similar across subjects, when viewing the same panoramic video.

Analysis: The HM scan-paths can be decomposed into magnitude and direction. Here, we measure the magnitude of HM scan-paths across different subjects. For each individual sequence in our PVS-HM database, Figure 3 plots the mean and standard deviation of HM scan-path magnitude over all 58 subjects. The last column of Figure 3 further shows the mean and standard deviation of HM scan-path magnitudes, averaged over all 76 sequences. We can see from this figure that the standard deviation is much less than the mean value, for all 76 sequences and for the overall results. Thus, we can conclude that there exists similarity for the magnitude of HM scan-paths across subjects, when the subjects watch the same panoramic sequence. This completes the analysis of *Finding 3*.

Finding 4: The direction of HM scan-paths on panoramic video is with high consistency across subjects.

Analysis: In our PVS-HM database, we evaluate the consistency of the HM scan-path direction among all 58 subjects. Specifically, we evaluate the direction of the scan-path starting from the consistent HM regions, since *Finding 1* has shown that the HM positions of different subjects are highly consistent. In our PVS-HM database, the consistent regions are extracted from 76 panoramic video sequences, which have HM positions of at least 12 subjects within a small great-circle distance range of 3° [41]. For each panoramic video sequence, Figure 4 shows the circular standard deviation [42] of HM scan-path directions, starting from the consistent HM

regions within one-second time slot². The last column of Figure 4 also reports the circular standard deviation averaged over all 76 sequences. We can see from this figure that the circular standard deviation of the HM scan-path direction is averagely 38.1° , and it is significantly less than 103.9° of randomly generated HM scan-path directions. Besides, the HM scan-path directions of all 76 sequences have considerably smaller circular standard deviation, compared to the random scan-paths. This implies that there exists high consistency on directions of HM scan-paths across subjects. Consequently, *Finding 4* can be validated.

Finding 5: Almost 50% subjects are consistent in one HM scan-path direction (among 8 uniform-quantized directions), and over 85% subjects are consistent in three directions for HM scan-paths.

Analysis: The distribution of HM scan-path directions in the PVS-HM database is analyzed as follows. Only HM scan-paths falling into consistent HM regions (mentioned in *Finding 4*) are selected for the analysis. Specifically, we discretize continuous $0 - 360^\circ$ directions of HM scan-paths by 8-level uniform quantization: $\{0^\circ, 45^\circ, 90^\circ \dots, 315^\circ\}$. Then, we count the proportions of subjects, whose HM scan-paths belong to the same discretized direction. Next, such proportions are ranked by their values in each extracted HM region, i.e., ranking from the 1-st to 8-th. For each ranking, the proportions of subjects are averaged over all HM regions for each panoramic sequence, which are shown in Figure 5. We can see from this figure that for all 76 sequences, the HM scan-paths of 50% subjects or more are consistent in the first ranked direction, and the HM scan-paths of over 85% subjects are consistent in top 3 directions. Figure 6 further shows

2. We have conducted our experiments with time slot being 0.1, 1 and 2 seconds, and the results are similar.

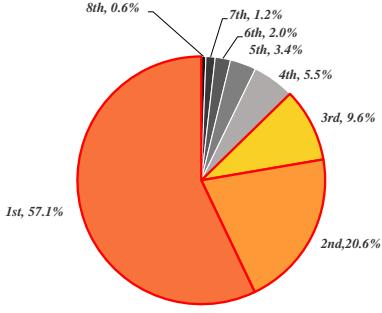


Fig. 6: Proportions of subjects with scan-path falling into each of 8 ranked directions. The proportions are averaged over all 76 sequences in the PVS-HM database.

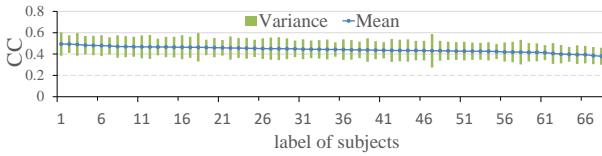


Fig. 7: Correlation coefficient of HM scan-paths over different time intervals and its variance.

the proportions of subjects with HM scan-path directions ranking from 1-st to 8-th, which are averaged over all 76 sequences in the PVS-HM database. As seen in this figure, the directions of HM scan-paths from 57.1%, 20.6% and 9.6% subjects belong to the top 3 ranked directions. In contrast, the HM scan-path directions of 12.7% subjects fall into other 5 directions. Therefore, *Finding 5* can be validated.

Finding 6: For one subject, the HM scan-path at the current time interval relies on that at the previous time interval and observed video content.

Analysis: For each individual subject, the vectors of HM scan-paths across the time interval of 0.3 second are extracted from all 76 sequences in our the PVS-HM database. Here, we calculate the CC values of HM scan-path vectors between two successive time periods, and then average the CC values over all 76 sequences for each subject. Figure 7 shows the averaged CC values for each of 58 subjects, which range from 0.39 - 0.51. This figure implies that the HM scan-path at the current time interval is somewhat correlated with that at the previous time interval. On the other hand, the CC value in Figure 7 is not sufficiently large, indicating that there exist other factors influencing HM scan-path. Actually, [18] shows that the current HM scan-path of one subject is also related to observed video content. The above completes the validation of *Finding 6*.

4 OFFLINE-DHP APPROACH

4.1 Framework of offline-DHP

In this section, we present our offline-DHP approach, in light of our findings in Section 3.2. Figure 8 shows the overall framework of our approach, in which the multiple DRL workflows are embedded to generate HM maps of the input panoramic video frames. The notations used in this figure and in our approach are listed in Table 2.

As shown in Figure 8, the input to our offline-DHP approach is the panoramic video frames $\{\mathbf{F}_t\}_1^T$. Since *Finding 1* has shown

that the HM positions are highly consistent across different subjects, we propose to generate HM maps for modeling human attention on the panoramic video, seen as the output of our offline-DHP approach. The HM map \mathbf{H}_t of frame t represents the probability of each pixel being the HM position. Similar to saliency maps of 2D videos, \mathbf{H}_t is obtained by convoluting the predicted HM positions $\{(\hat{x}_t^n, \hat{y}_t^n)\}_{n=1}^N$ with a 2D Gaussian filter. Because *Finding 5* has found that the HM scan-paths of different subjects are consistent in more than one directions, the HM positions $\{(x_t^m, y_t^m)\}_{m=1}^M$ of M subjects may be different from each other. Accordingly, this paper assumes that the number of the predicted HM positions N is equivalent to M at each frame, for predicting HM positions of all subjects. In other words, to obtain $(\hat{x}_t^n, \hat{y}_t^n)$, our offline-DHP approach estimates the HM positions of one subject through HM scan-path prediction of one DRL workflow. Then, N DRL workflows are run to obtain N HM positions $\{(\hat{x}_t^n, \hat{y}_t^n)\}_{n=1}^N$ at frame t , corresponding to the ground-truth HM positions $\{(x_t^m, y_t^m)\}_{m=1}^M$ of M ($= N$) subjects at this frame. Each of DRL workflows works independently to generate a HM scan-path by randomly sampling actions based on a learnt policy π_t . Note that all DRL workflows share the same policy π_t in our approach.

In a single DRL workflow, $\{(\hat{x}_t^n, \hat{y}_t^n)\}_{t=1}^T$ can be modeled by determining a series of actions: $\{\hat{\alpha}_t^n\}_{t=1}^T$ and $\{\hat{\nu}_t^n\}_{t=1}^T$. It is worth pointing out that $\{\hat{\alpha}_t^n\}_{t=1}^T$ and $\{\hat{\nu}_t^n\}_{t=1}^T$, as the actions in the DRL workflow, are predictable, since *Findings 3* and *4* have indicated that subjects are consistent in the magnitudes and directions of HM scan-paths. As can be seen in Figure 8, in each workflow, one HM scan-path is generated through the interaction between FoV extractor and HM scan-path predictor. Note that the extracted FoV is $103^\circ \times 60^\circ$, the same as the setting of the HMD. Specifically, Figure 8 shows that FoV \mathbf{o}_t^n is extracted via making its center locate at the HM position $(\hat{x}_t^n, \hat{y}_t^n)$, and $(\hat{x}_t^n, \hat{y}_t^n)$ is generated by the predicted action of HM scan-paths $(\hat{\alpha}_{t-1}^n, \hat{\nu}_{t-1}^n)$ at the previous video frame. Then, the content of extracted FoV works as observation of DRL, for predicting the next action of HM scan-path $(\hat{\alpha}_t^n, \hat{\nu}_t^n)$. The HM scan-path generated by each DRL workflow is forwarded to obtain HM positions at incoming frames. Afterwards, the HM positions from multiple DRL workflows are integrated, and then smoothed by the 2D Gaussian filter. Finally, the HM maps $\{\mathbf{H}_t\}_1^T$ of the panoramic video are obtained, which model the heat maps for HM positions at each frame.

4.2 DRL model of the offline-DHP approach

As described in Sections 4.1, the DRL workflow is a key component in our offline-DHP framework, which targets at predicting HM scan-paths. This section presents how to train the DRL model of each workflow in Figure 8 for predicting HM maps. In this section, we take the n -th workflow as an example. Figure 9 shows the framework of training the DRL model. As shown in Figure 9, the FoV of input video frame is extracted upon the action of the predicted HM scan-path at the previous frame. The extracted FoV, as the observation, is then feeded into the DRL network. In addition, the reward, which measures the similarity between the prediction and ground-truth of HM scan-paths, is estimated to evaluate the action made by the DRL model. Then, the reward is used to make decision on the action through the DRL model, i.e., the HM scan-path at the current frame. In the DRL model, the observation of the extracted FoV and the reward of HM scan-path prediction compose an environment.

In training the DRL model, environment interacts with the HM scan-path predictor. The interaction is achieved in our DRL model

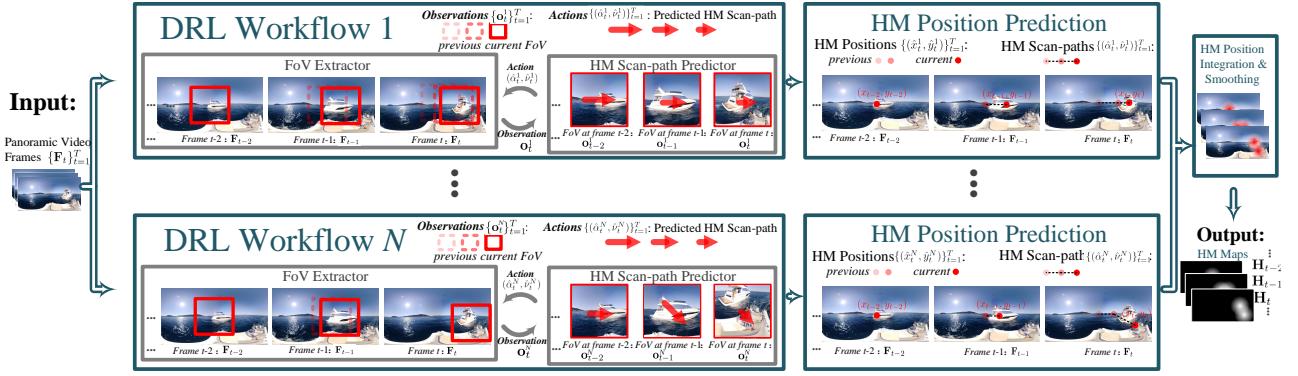


Fig. 8: Framework of our Offline-DHP approach.

TABLE 2: Notations denoted in Figure 8

$\{\mathbf{F}_t\}_{t=1}^T$	The panoramic frames with frame number t ranging from 1 to T , as the input to offline-DHP
$\{\mathbf{H}_t\}_{t=1}^T$	The HM maps for frames from 1 to T , as the output of offline-DHP
$\{m\}_{m=1}^M$	The subjects from 1 to M , with m being the m -th subject
$\{n\}_{n=1}^N$	The DRL workflows from 1 to N , with n being the n -th workflow
$\{\mathbf{o}_t^n\}_{t=1}^T$	The FoV for frames from 1 to T , as the <i>observation</i> of the n -th DRL workflow
(x_t^n, y_t^n)	The ground-truth HM position of the m -th subject at frame t
$(\hat{x}_t^n, \hat{y}_t^n)$	The HM position predicted by the n -th DRL workflow at frame t
π_t	The predicted probability distribution of HM direction at frame t , as the <i>policy</i> of DRL
$\hat{\alpha}_t^n$	The predicted HM scan-path direction at frame t from the n -th DRL workflow, as the <i>action</i> of DRL
α_t^m	The ground-truth HM scan-path direction of the m -th subject at frame t
$\hat{\nu}_t^n$	The predicted HM scan-path magnitude at frame t from the n -th DRL workflow, as the <i>action</i> of DRL
ν_t^m	The ground-truth HM scan-path magnitude of the m -th subject at frame t
$r_{n,t}^\alpha$	The <i>reward</i> for deciding $\hat{\alpha}_t^n$ in the n -th DRL workflow
$r_{n,t}^\nu$	The <i>reward</i> for deciding $\hat{\nu}_t^n$ in the n -th DRL workflow
\mathbf{f}_t^n	The extracted LSTM feature at frame t , as part of <i>observed state</i> in the n -th DRL workflow

by the following procedure.

- (1) At frame t , the FoV extractor obtains the current *observation* \mathbf{o}_t^n ($103^\circ \times 60^\circ$) from the input video frame \mathbf{F}_t , according to the predicted HM position $(\hat{x}_t^n, \hat{y}_t^n)$. In our work, \mathbf{o}_t^n is projected to the 2D region and then down-sampled to 42×42 .
- (2) The current \mathbf{o}_t^n and the LSTM feature \mathbf{f}_{t-1}^n from the last frame are delivered to the DRL network in the HM scan-path predictor. In our work, the DRL network contains four convolutional layers [43] and one LSTM layer [44], [45], which are used to extract the spatial and temporal features, respectively. The details about the architecture of the DRL network can be found in Figure 9.
- (3) At frame t , the DRL network produces the LSTM feature \mathbf{f}_t^n , HM scan-path magnitude $\hat{\nu}_t^n$ and policy π_t . Here, π_t is the probability distribution over the *actions* of HM scan-path directions.
- (4) Given π_t , the HM scan-path predictor randomly samples an *action* $\hat{\alpha}_t^n$, with standard deviation ε to ensure exploration. Here, $\hat{\alpha}_t^n$ includes 8 discrete directions in GDS: $\{0^\circ, 45^\circ, \dots, 315^\circ\}$.
- (5) *Environment* is updated with $\hat{\nu}_t^n$ and $\hat{\alpha}_t^n$, leading to $(\hat{x}_t^n, \hat{y}_t^n) \rightarrow (\hat{x}_{t+1}^n, \hat{y}_{t+1}^n)$. The FoV extractor returns a new \mathbf{o}_{t+1}^n according to the HM position $(\hat{x}_{t+1}^n, \hat{y}_{t+1}^n)$. The *reward* estimator returns the *rewards* $r_{n,t}^\nu$ and $r_{n,t}^\alpha$ in predicting $\hat{\nu}_t^n$ and $\hat{\alpha}_t^n$, upon the ground-truth HM scan-paths of $\{\nu_t^m\}_{m=1}^M$ and $\{\alpha_t^m\}_{m=1}^M$.
- (6) A set of experiences $\{\mathbf{o}_t^n, \mathbf{f}_{t-1}^n, \hat{\nu}_t^n, \hat{\alpha}_t^n, r_{n,t}^\nu, r_{n,t}^\alpha\}$ are stored in an experience buffer for frame t . In addition, \mathbf{o}_{t+1}^n and \mathbf{f}_t^n are preserved for processing frame $t+1$.

- (7) Once t meets the terminal condition of exceeding the frame number T , all experiences in the buffer are delivered to the optimizer for updating the DRL network.

Reward Estimation. Next, we focus on modeling the *rewards*: $r_{n,t}^\alpha$ and $r_{n,t}^\nu$. When training the DRL model, our goal is to make $\hat{\alpha}_t^n$ and $\hat{\nu}_t^n$ approach to the ground-truth HM data. Thus, the *rewards* $r_{n,t}^\alpha$ and $r_{n,t}^\nu$ can be represented by the differences from $\hat{\alpha}_t^n$ and $\hat{\nu}_t^n$ to $\{\alpha_t^m\}_{m=1}^M$ and $\{\nu_t^m\}_{m=1}^M$, respectively. In our approach, we use the Gaussian distribution to measure the differences between $\hat{\alpha}_t^n$ and the ground-truth $\{\alpha_t^m\}_{m=1}^M$. In addition, we also consider the distances from predicted HM position $(\hat{x}_t^n, \hat{y}_t^n)$ to $\{(x_t^m, y_t^m)\}_{m=1}^M$ in calculating the *rewards* of $r_{n,t}^\alpha$ and $r_{n,t}^\nu$. Such distances are modeled by the 2D Gaussian distribution. Then, $r_{n,t}^\alpha$ can be written as

$$r_{n,t}^\alpha = \frac{1}{N} \sum_{m=1}^M e^{-\frac{1}{2} \left(\frac{D_d(\hat{\alpha}_t^n, \alpha_t^m)}{\rho} \right)^2} e^{-\frac{1}{2} \left(\frac{D_s((\hat{x}_t^n, \hat{y}_t^n), (x_t^m, y_t^m))}{\varrho} \right)^2}, \quad (1)$$

In (1), D_d defines the *phase difference*, and D_s denotes the *great-circle distance* [46]. Moreover, ρ and ϱ are the standard deviations of Gaussian distributions, as the hyper-parameters. Similarly, we have

$$r_{n,t}^\nu = \frac{1}{N} \sum_{m=1}^M e^{-\frac{1}{2} \left(\frac{\hat{\nu}_t^n - \nu_t^m}{\varsigma} \right)^2} e^{-\frac{1}{2} \left(\frac{D_d(\hat{\alpha}_t^n, \alpha_t^m)}{\rho} \right)^2} e^{-\frac{1}{2} \left(\frac{D_s((\hat{x}_t^n, \hat{y}_t^n), (x_t^m, y_t^m))}{\varrho} \right)^2}. \quad (2)$$

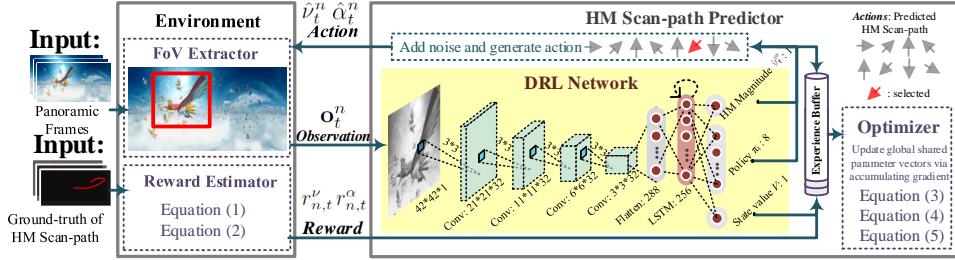


Fig. 9: Framework of training the DRL model for obtain each DRL workflow in Figure 8.

where ς is the hyper-parameter for the standard deviation of HM scan-path magnitude.

Optimization. Next, we need to optimize the rewards: $r_{n,t}^\alpha$ and $r_{n,t}^\nu$, when learning the network parameters of our DRL model in Figure 9. Our offline-DHP approach implements the asynchronous DRL method [9] to learn the DRL parameters with optimized rewards. Hence, multiple workflows are run to interact with multiple environments with workflow-specific parameter vectors $\{\theta_\nu^n, \theta_\pi^n, \theta_V^n\}$, producing \hat{v}_t^n , $\hat{\alpha}_t^n$ and V . Here, V denotes state value output by the DRL network, which is obtained using the same way as [9]. Meanwhile, global-shared parameter vectors $\{\theta_\nu, \theta_\pi, \theta_V\}$ ³ are updated via accumulating gradient. For more details about the workflow-specific and global-shared parameter vectors, refer to [9]. In our approach, reward $r_{n,t}^\nu$ is optimized to train θ_ν as follows,

$$d\theta_\nu \leftarrow d\theta_\nu + \nabla_{\theta_\nu^n} \sum_{t=1}^T r_{n,t}^\nu. \quad (3)$$

Besides, we can optimize reward $r_{n,t}^\alpha$ by

$$d\theta_V \leftarrow d\theta_V + \nabla_{\theta_V^n} \sum_{t=1}^T \left(\sum_{i=t}^T \gamma^{i-t} r_{n,i}^\alpha - V(\mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta_V^n) \right)^2, \quad (4)$$

$$\begin{aligned} d\theta_\pi \leftarrow d\theta_\pi + \nabla_{\theta_\pi^n} & \sum_{t=1}^T \log \pi(\hat{\alpha}_t^n | \mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta_\pi^n) \cdot \\ & \left(\sum_{i=t}^T \gamma^{i-t} r_{n,i}^\alpha - V(\mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta_V^n) \right), \end{aligned} \quad (5)$$

where γ is the discount factor of *Q-learning* [47]. In addition, $V(\mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta_V^n)$ denotes state value V yielded by \mathbf{o}_t^n , \mathbf{f}_{t-1}^n and θ_V^n ; $\pi(\hat{\alpha}_t^n | \mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta_\pi^n)$ stands for the probability of action $\hat{\alpha}_t^n$ that is yielded by policy π_t from \mathbf{o}_t^n , \mathbf{f}_{t-1}^n and θ_π^n . Finally, upon the above equations, RMSProp [48] is applied to optimize rewards in training data. As a result, the workflow-specific and global-shared parameter vectors can be learned to predict HM scan-paths.

5 ONLINE-DHP APPROACH

In this section, we present our online-DHP approach. The online-DHP approach refers to predicting the a specific subject's HM position $(\hat{x}_{t+1}, \hat{y}_{t+1})$ at frame $t+1$, given his/her HM positions $\{(x_1, y_1), \dots, (x_t, y_t)\}$ till frame t . Note that the definition of

3. As can be seen in Figure 9, $\{\theta_\nu, \theta_\pi, \theta_V\}$ share all CNN and LSTM layers in our offline-DHP approach, but they are separated at the output layer.

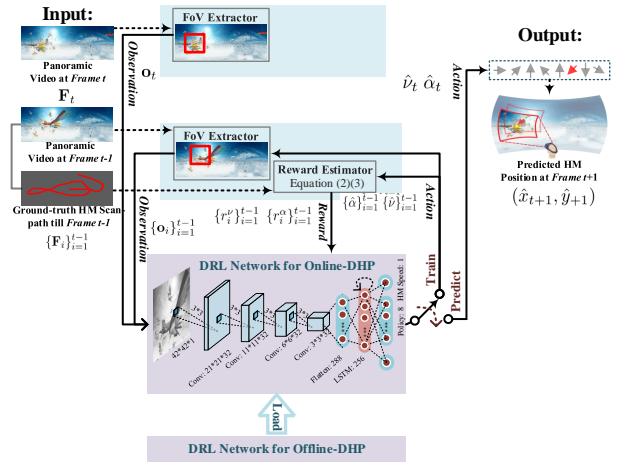


Fig. 10: Framework of our online-DHP approach.

notations in this section is similar to those of Section 4, and the only difference is that n is removed in all notations as there is only one subject/workflow in online-DHP. In this section, we define the subject as the *viewer*, whose HM positions need to be predicted online. Figure 10 shows the framework of our online-DHP approach. According to *Finding 6*, the input to our online-DHP framework is the *viewer's* HM scan-path $\{(\alpha_1, \nu_1), \dots, (\alpha_{t-1}, \nu_{t-1})\}$ and frame content $\{\mathbf{F}_1, \dots, \mathbf{F}_t\}$, while the output is the predicted HM position $(\hat{x}_{t+1}, \hat{y}_{t+1})$ at the next frame for the *viewer*. This can be seen as online prediction of HM positions $\{(\hat{x}_t, \hat{y}_t)\}_{t=1}^T$. To this end, our online-DHP consists of two stages: the training and prediction stages. At the first stage, the parameters of the DRL network are trained. At the second state, the *action* of HM scan-path is yielded upon the trained DRL network, to predict the HM position online. In the following, we discuss these two stages with more details.

5.1 Stage I: Training

At the beginning frame, the HM position (\hat{x}_1, \hat{y}_1) of the *viewer* is initialized to be the center of front region, which is the general setting of the panoramic video player. Then, the trained DRL network of our offline-DHP approach is loaded as the initial DRL network for online prediction, and both networks share the

same structure. The reason for loading the offline-DHP network is that it contains the knowledge of HM related features. Later, this initial DRL network is fine-tuned by the *viewer's* HM scan-path at incoming frames.

Next, we focus on the algorithm of training the DRL network in our online-DHP approach. As aforementioned, the initial parameters of the DRL network at the first frame are directly from those of offline-DHP. At each of the incoming frames, several episodes are run to update the DRL network of online-DHP. The following summarizes the procedure of one episode at frame $t + 1$.

- 1) Iterate the following steps from $i = 1$ to t . At each iteration, $(\hat{\alpha}_i, \hat{\nu}_i)$ and (α_i, ν_i) are the predicted and ground-truth *actions* of the HM scan-path for the subject, and \mathbf{o}_i is the *observation* of the FoV content.
- 2) Take *action* of $(\hat{\alpha}_i, \hat{\nu}_i)$ using the DRL network, given the current *observation* $\{\mathbf{o}_1, \dots, \mathbf{o}_i\}$ till frame i . The *action* of $\hat{\alpha}_i$ selects 1 among 8 discrete HM scan-path directions, i.e., $\{0^\circ, 45^\circ, \dots, 315^\circ\}$. The *action* of $\hat{\nu}_i$ also contains a scalar of HM scan-path magnitude.
- 3) Calculate *rewards* (r_i^α, r_i^ν) from the *reward estimator* with (1) and (2), which measures how close the *action* $(\hat{\alpha}_i, \hat{\nu}_i)$ to the ground-truth HM scan-path (α_i, ν_i) of the *viewer*. Here, the *reward* calculation does not require the sums of (1) and (2), since the ground-truth HM scan-path of online prediction is from a single *viewer*, rather than from all subjects.
- 4) Generate new *observation* \mathbf{o}_{i+1} from the FoV extractor with the above *action* $(\hat{\alpha}_i, \hat{\nu}_i)$, and then it is input to the DRL network.
- 5) Update the DRL network using (3), (4) and (5), if the iteration number i is equivalent to t .

Here, the definitions of *action*, *reward* and *observation* are the same as Section 4.2. The above iterations share the same implementation of training the DRL model in offline-DHP, which is already presented in Section 4.2.

Once the above iterations are terminated, our algorithm moves to the next episode. After a number of episodes, the training stage ends for frame $t + 1$, when meeting the the termination conditions. In our approach, there are two criteria for the termination condition. The first criterion is the maximum number E of episodes, which is set to be 500 in this paper. The second one is based on the metric of mean overlap (MO), which measures how close the predicted HM position to the ground-truth HM position. MO ranges from 0 to 1, and larger MO indicates more precise prediction. Specifically, MO is defined as,

$$MO = \frac{A(\text{FoV}_p \cap \text{FoV}_g)}{A(\text{FoV}_p \cup \text{FoV}_g)}, \quad (6)$$

where FoV_p and FoV_g represent the FoVs at predicted and ground-truth HM positions, respectively. In (6), A represents the area of a panoramic region, which counts for number of pixels.

Finally, the training DRL network can be obtained at frame $t + 1$, once satisfying one of the above termination criteria. Algorithm 1 presents the summary of the training stage in online-DHP.

5.2 Stage II: Prediction

When the average MO is larger than a threshold th_{mo} , the switch in Figure 10 is turned to “predict” and the DRL network make a prediction for the action of HM scan-path at frame $t + 1$. Note that

Algorithm 1 The algorithm of the training stage in online-DHP for frame $t + 1$.

Input: Panoramic video frames $\{\mathbf{F}_1, \dots, \mathbf{F}_t\}$, and the ground-truth HM positions of the *viewer* $\{(x_1, y_1), \dots, (x_t, y_t)\}$;
 Initialize the DRL network of online-DHP with parameter vectors $\{\theta_\nu, \theta_\pi, \theta_V\}$, by loading the network of offline-DHP;
for $e = 1$ **to** E **do**
 Initialize the HM position to be center of front region: $\hat{x}_1 = 0, \hat{y}_1 = 0$;
 Initialize LSTM feature to be the zero vector: $\mathbf{f}_0 = \mathbf{0}$;
 for $i = 1$ **to** $t - 1$ **do**
 Extract *observation* \mathbf{o}_i (FoV) from \mathbf{F}_i according to (\hat{x}_i, \hat{y}_i) ;
 Obtain *policy* π_i and LSTM feature \mathbf{f}_i using the DRL network with $\{\mathbf{o}_i, \mathbf{f}_{i-1}, \theta_\pi\}$;
 Select *action* $\hat{\alpha}_i$ according to the ϵ -greedy policy of π_i ;
 Generate *action* $\hat{\nu}_i$ using the DRL network given $\mathbf{o}_i, \mathbf{f}_{i-1}$ and θ_ν ;
 Calculate $(\hat{x}_{i+1}, \hat{y}_{i+1})$ with regard to $\hat{\alpha}_i, \hat{\nu}_i$, and (\hat{x}_i, \hat{y}_i) ;
 Estimate *rewards* r_i^ν, r_i^α of $\hat{\alpha}_i, \hat{\nu}_i$ using (1), (2);
 Calculate MO between (\hat{x}_i, \hat{y}_i) and (x_i, y_i) , denoted as MO_i ;
 Store a set of experience: $\{\mathbf{o}_i, \mathbf{f}_{i-1}, \hat{\nu}_i, \hat{\alpha}_i, r_i^\nu, r_i^\alpha\}$;
 $i \leftarrow i + 1$;
 end for
 Update $\{\theta_\nu, \theta_\pi, \theta_V\}$ according to (3), (4), (5), in which $\{\theta_\nu^n, \theta_\pi^n, \theta_V^n\}$ are replaced by $\{\theta_\nu, \theta_\pi, \theta_V\}$;
 $e \leftarrow e + 1$;
 $MO = \frac{\sum_{i=1}^{t-1} MO_i}{t-1}$;
 if $MO > th_{MO}$ **then**
 break;
 end if
 end for
Return: The trained parameter vectors: $\{\theta_\nu, \theta_\pi, \theta_V\}$.

if the number of training episodes exceeds E , the “predict” is also switched on, so that the training episodes end in a limited time. When entering the prediction stage, the DRL model trained at the first stage is used to produce the HM position as follows.

First, the LSTM features $\{\mathbf{f}_i\}_{i=1}^{t-1}$ are sequentially updated from frame 1 to $t - 1$, upon the observed FoVs $\{\mathbf{o}\}_{i=1}^{t-1}$ and the DRL parameters θ_π of the training stage. Note that the LSTM feature is initialized with the zero vector $\mathbf{0}$ at frame 1. Then, $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_\pi\}$ produces the action $\hat{\alpha}_t$ of HM scan-path direction. In addition, the HM scan-path magnitude $\hat{\nu}_t$ is generated using $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_\nu\}$, in which the parameters of θ_ν are obtained at the training stage. Afterwards, the HM position $(\hat{x}_{t+1}, \hat{y}_{t+1})$ at frame $t + 1$ can be predicted, given the ground-truth HM position (x_t, y_t) and the estimated HM scan-path $(\hat{\alpha}_t, \hat{\nu}_t)$ at frame t . Algorithm 2 presents the summary of the prediction stage in online-DHP. Finally, online-DHP is achieved by alternating between the stages of training and prediction till the last frame.

Algorithm 2 The algorithm of the prediction stage in online-DHP for frame $t + 1$.

Input: The trained parameter vectors: $\{\theta_\nu, \theta_\pi, \theta_V\}$ from the training stage, panoramic video frames $\{\mathbf{F}_1, \dots, \mathbf{F}_t\}$, and the ground-truth HM positions of the *viewer* $\{(x_1, y_1), \dots, (x_t, y_t)\}$;
 Initialize the LSTM feature with the zero vector: $\mathbf{f}_0 = \mathbf{0}$;
for $i = 1$ **to** $t - 1$ **do**
 Extract *observation* \mathbf{o}_i (FoV) from \mathbf{F}_i according to (x_i, y_i) ;
 Obtain LSTM feature \mathbf{f}_i using the DRL network with $\{\mathbf{o}_i, \mathbf{f}_{i-1}, \theta_\pi\}$;
 $i \leftarrow i + 1$;
 end for
 Extract *observation* \mathbf{o}_t (FoV) from \mathbf{F}_t according to (x_t, y_t) ;
 Obtain *policy* π_t using the DRL network with $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_\pi\}$;
 Choose *action* $\hat{\alpha}_t$ with the greedy policy based on π_t ;
 Generate HM magnitude $\hat{\nu}_t$ using the DRL network with $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_\nu\}$;
 Estimate HM position $(\hat{x}_{t+1}, \hat{y}_{t+1})$ at frame $t + 1$, upon $\hat{\alpha}_t, \hat{\nu}_t$ and (x_t, y_t) ;
Return: The HM position at frame $t + 1$: $(\hat{x}_{t+1}, \hat{y}_{t+1})$.

TABLE 3: CC results of HM map prediction by our and other approaches

CC	Method	StarryPolar	Symphony	SpaceWar	RioOlympics	InsideCar	SpaceWar2	Sunset	BlueWorld	Waterfall	Dancing	CMLauncher2	Guitar	KingKong	BTSRun	WaitingForLove	Average
Non-FCB	Our	0.185	0.710	0.573	0.717	0.783	0.673	0.673	0.678	0.763	0.837	0.585	0.645	0.751	0.764	0.471	0.654
	BMS	0.450	0.167	0.274	0.228	0.331	0.067	0.463	0.169	0.393	0.121	0.203	0.328	0.105	0.105	0.223	0.242
	OBDL	0.107	0.184	0.028	0.190	0.260	0.100	0.308	0.027	0.025	0.176	0.117	0.066	0.125	0.047	0.222	0.132
	SALICON	0.293	0.129	0.126	0.153	0.364	0.034	0.186	0.265	0.103	0.120	0.166	0.216	0.150	0.063	0.256	0.175
FCB	Our	0.497	0.816	0.574	0.768	0.712	0.655	0.810	0.748	0.797	0.764	0.747	0.652	0.673	0.679	0.677	0.704
	BMS	0.692	0.567	0.520	0.494	0.495	0.368	0.711	0.500	0.655	0.414	0.546	0.494	0.311	0.322	0.503	0.506
	OBDL	0.510	0.540	0.321	0.441	0.496	0.455	0.638	0.464	0.434	0.408	0.468	0.461	0.410	0.288	0.598	0.462
	SALICON	0.664	0.563	0.456	0.539	0.528	0.452	0.658	0.596	0.525	0.355	0.667	0.461	0.362	0.346	0.628	0.520
FCB Only		0.557	0.747	0.317	0.403	0.292	0.239	0.585	0.477	0.583	0.387	0.735	0.356	0.271	0.201	0.497	0.443

TABLE 4: NSS results of HM map prediction by our and other approaches

NSS	Method	StarryPolar	RioOlympics	SpaceWar2	Symphony	SpaceWar	Waterfall	Sunset	BlueWorld	Guitar	Dancing	InsideCar	CMLauncher2	WaitingForLove	BTSRun	KingKong	Average
Non-FCB	Our	0.899	2.806	2.237	3.346	2.180	3.765	2.529	3.196	3.461	5.297	4.402	3.529	2.278	4.572	3.334	3.189
	BMS	1.313	0.772	0.137	0.710	0.807	1.673	1.613	0.841	1.497	0.670	1.657	1.034	0.997	0.546	0.119	0.959
	OBDL	0.126	0.637	0.301	0.260	0.064	0.073	1.015	0.035	0.393	0.980	1.375	0.660	0.964	0.215	0.107	0.480
	SALICON	0.730	0.600	0.251	0.456	0.344	0.410	0.669	1.138	0.965	0.230	1.823	0.921	1.298	0.337	0.203	0.692
FCB	Our	1.825	2.911	2.064	3.756	2.031	3.755	2.943	3.393	3.395	4.608	3.816	4.463	3.351	3.931	2.883	3.275
	BMS	2.206	1.779	1.063	2.537	1.667	2.891	2.507	2.280	2.386	2.366	2.508	3.136	2.434	1.771	1.288	2.188
	OBDL	1.712	1.572	1.371	2.368	1.055	1.920	2.225	2.007	2.377	2.319	2.556	2.777	2.912	1.580	1.693	2.030
	SALICON	2.083	2.024	1.332	2.477	1.493	2.353	2.352	2.619	2.264	1.957	2.672	3.932	3.143	1.915	1.496	2.274
FCB Only		2.388	1.613	0.699	4.123	1.190	3.191	2.406	2.286	1.828	2.151	1.387	5.764	2.600	1.095	1.020	2.249

6 EXPERIMENTAL RESULTS

This section presents experimental results to validate the effectiveness of our offline-DHP and online-DHP approaches. In Section 6.1, we discuss the settings of offline-DHP and online-DHP in our experiments. Sections 6.2 and 6.3 compares the performance of our offline-DHP and online-DHP approaches with other approaches, in predicting HM positions of offline and online scenarios, respectively.

6.1 Settings

For evaluating the performance of offline-DHP, we randomly divided all 76 panoramic video sequences of our PVS-HM database into a training set (61 sequences) and a test set (15 sequences). In training the DRL model, the hyperparameters ρ and ϱ of (1) and (2) were tuned over the test set, to estimate *reward* of HM scan-path prediction. As a result, ρ and ϱ were set to be 42 and 0.7. In addition, we followed [9] to set other hyperparameters of DRL. For example, we chose the discount factor γ of (4) and (5) to be 0.99 for *reward* optimization. In our experiments, all 61 training sequences, each of which corresponds to a local DRL network, were used to update the global network as the trained DRL model. The number of DRL workflows N in the offline-DHP framework was set to be 56, the same as the number of subjects in our PVS-HM database. Similar to [41], at each panoramic frame the HM positions predicted by 56 DRL workflows were convoluted with a 2D Gaussian filter, in order to generate the HM map. In our experiments, the HM maps in a panorama were projected to the 2D coordination for facilitating visualization.

For evaluating the performance of online-DHP, XXX

6.2 Performance evaluation on offline-DHP

Now, we evaluate the performance of our offline-DHP approach in predicting HM maps of all 15 test sequences from the PVS-HM database. To the best of our knowledge, there is no work on predicting HM maps of panoramic video, and saliency prediction is the closest field. Therefore, we compare our offline-DHP approach

with three state-of-the-art saliency detection approaches, i.e., OBDL [25], BMS [13] and SALICON [14]. In particular, OBDL [25] and BMS [13] are the latest saliency detection approaches for video and image, respectively. SALICON [14] is a state-of-the-art DNN approach for saliency detection. In addition to the above three approaches, we also compare our approach to the front-center-bias (FCB) baseline, since human attention normally biases towards front-center regions of panoramic video. Here, we model FCB by Gaussian distribution⁴, similar to the center bias of saliency detection. In the field of saliency detection, the center bias [2] is normally considered in saliency maps to improve saliency detection accuracy. Hence, we further report results of HM maps combined with the FCB feature, in our and other approaches. See Appendix A for more details about the combination of FCB. For evaluation, we measure the accuracy of HM map prediction in terms of CC and normalized scan-path saliency (NSS), which are two effective evaluation metrics [?] in saliency detection. Note that the larger value of CC or NSS means more accurate prediction of HM maps.

Table 5 tabulates the results of CC and NSS in predicting HM maps of 15 test sequences, for our and other approaches. In this table, the results of CC and NSS are averaged over all frames for each test sequence. As shown in this table, when FCB is not integrated, our offline-DHP approach performs the best among all four approaches and the FCB baseline in terms of CC and NSS. More importantly, once integrated with FCB, all four approaches have performance improvement, and our approach still performs best among all three approaches. Specifically, our offline-DHP approach increases the CC value (averaged over all 15 test sequences) by 0.198, 0.242 and 0.184, compared with OBDL, BMS and SALICON, respectively. Additionally, the increase of averaged NSS value in our approach is 1.087, 1.245 and 1.001, over OBDL, BMS and SALICON. In a word, our offline-DHP approach is effective in predicting HM maps of panoramic video, much better than other approaches and the FCB baseline.

4. The standard deviation is 12.8 degree, obtained by least square fitting over all HM positions of the training set.

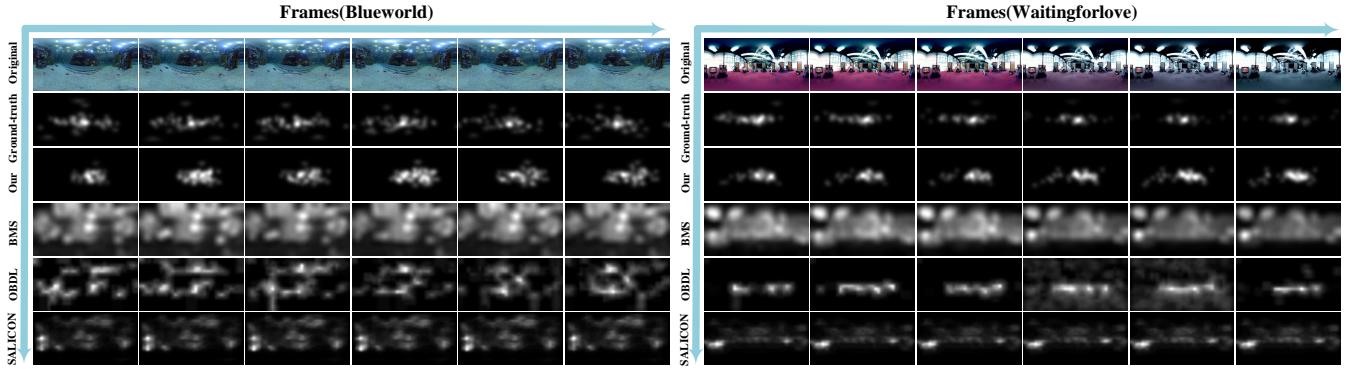


Fig. 11: HM maps of several frames selected from two test sequences in our PVS-HM database. They are all visualized in the 2D coordination. The second row shows the ground-truth HM maps, which are generated upon HM portions of all 56 subjects. The third to sixth rows show the HM maps of our, BMS [13], OBBL [25], and SALICON approaches. The last row demonstrates the HM maps by the FCB baseline.

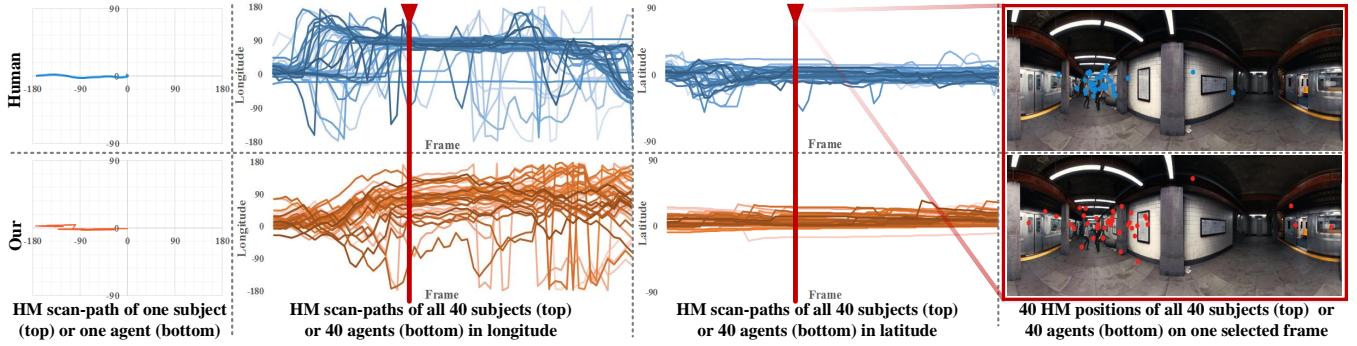


Fig. 12: Visualization in scan-paths and positions of HM generated by ground-truth and our offline-DHP approach, for sequence *Help*. The first column shows scan-path of one subject and one DRL workflow. The second and third columns show scan-paths of all 40 subjects and 40 DRL workflows, in longitude and latitude directions. The last column visualizes HM positions of ground-truth and our approach.

Next, we move to the comparison of subjective results. Figure 11 shows several frames from two selected sequences and their ground-truth HM maps of 58 subjects. In Figure 11, we further visualize the HM maps generated by our and other approaches including the FCB baseline. Here, the predicted HM maps are integrated with FCB, since the FCB feature can improve the performance of all four approaches in predicting HM maps (as presented in Table 5). From this figure, one can observe that the HM maps of our approach are more close to the ground-truth HM maps, compared to other approaches. This indicates that our offline-DHP approach is capable of better locating the HM positions of different subjects on panoramic video. Moreover, Figure 12 plots the HM scan-paths by different subjects and by multiple DRL workflows of offline-DHP, to investigate the agreement between the subjects and DRL workflows. We can see from this figure that the DRL workflows are able to yield similar scan-paths as humans. We further show in Figure 12 the HM positions obtained from the ground-truth and predicted HM scan-paths. This figure reveals that HM positions can be well predicted by our offline-DHP approach. In conclusion, our offline-DHP approach is effective in modeling HM maps by predicting both scan-paths and positions of HM for panoramic video.

6.3 Performance evaluation on online-DHP

We evaluate the performance of our online-DHP approach for predicting HM positions in the online scenario. The online scenario refers to predicting HM position of one subject at each frame upon the observed HM positions of this subject at the previous frames. In our experiments, we compare the performance of online-DHP with the state-of-the-art deep 360 pilot [18], which is the only existing approach for the online prediction of HM positions. We also compare our online-DHP with two baselines. According to *Finding 6*, the first baseline (called baseline 1) keeps the HM scan-path of the current time interval the same as that at the previous time interval, such that the HM position at each frame can be generated. The second baseline (called baseline 2) produces the HM position, using the randomly generated HM scan-path. The same as [18], MO of (6) is measured as the metric to evaluate the performance of online prediction for HM positions.

Table ?? compares the MO results of our and other approaches for the 15 test sequences of our PVS-HM database. Note that the MO results of each sequence are averaged over the predicted HM positions of all 58 subjects in our database. As observed in this table, our online-DHP approach is significantly superior to two baselines, indicating the effectiveness of applying DRL in predicting HM positions online. In addition, we can see from Table ?? that our online-DHP approach performs better than the deep 360 pilot [18], with an increase of XXXX in average MO. Such

TABLE 5: MO results of HM position prediction by our and other approaches

Method	KingKong	SpaceWar2	StarryPolar	Dancing	Guitar	BTSRun	InsideCar	RioOlympics	SpaceWar	CMLauncher2	Waterfall	Sunset	BlueWorld	Symphony	WaitingForLove	Average
Online	0.809	0.763	0.549	0.859	0.785	0.878	0.847	0.820	0.626	0.763	0.667	0.659	0.693	0.747	0.836	0.753
Deep 360 Pilot	0.266	0.285	0.258	0.386	0.341	0.381	0.258	0.240	0.298	0.270	0.274	0.283	0.314	0.432	0.393	0.312
Random-baseline	0.201	0.206	0.161	0.216	0.203	0.206	0.216	0.203	0.209	0.205	0.203	0.204	0.206	0.202	0.211	0.204
Keep-baseline	0.224	0.231	0.197	0.217	0.227	0.237	0.234	0.225	0.216	0.251	0.251	0.209	0.229	0.216	0.225	0.226

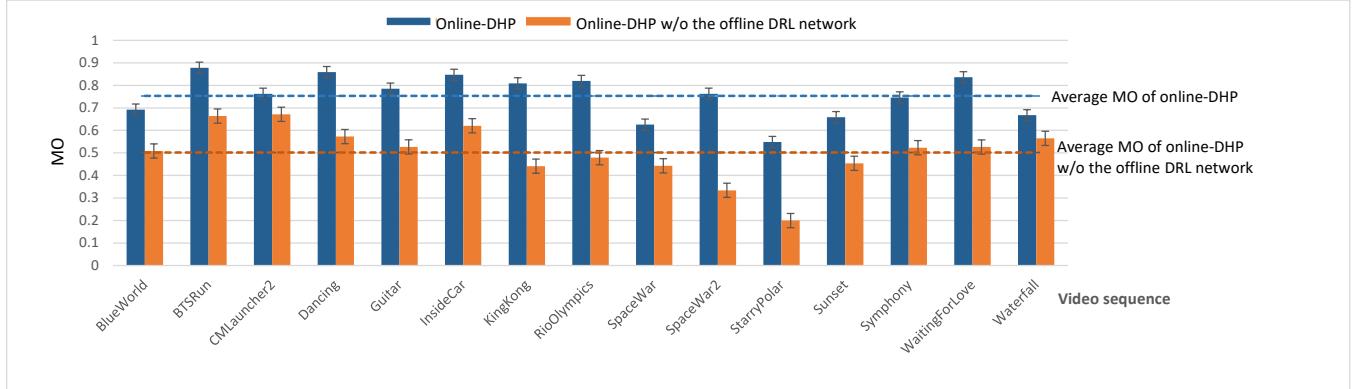


Fig. 13: MO results between Online-DHP and Online-DHP without offline-DHP.

a increase is probably because (1) the online DRL model of our approach is capable of generating the accurate *actions* of HM scan-paths. and (2) the DRL network of offline-DHP is incorporated in our online prediction as the prior knowledge.

To test the generalizability of our approach, we further evaluate the performance of our, the deep 360 pilot [18] and two baseline approaches on the test set of [18]. For such an evaluation, our online-DHP is still based on our offline DRL network that is learned from the training sequences of our PVS-HM database. The MO results are shown in Table XXX. From this table, we can see that our online-DHP approach again outperforms [18] and two baselines. In particular, our approach has XXX MO increase over [18], despite testing on the test set of [18]. Therefore, the generalization capability of our approach can be confirmed.

It is interesting to analyze the gain of incorporating the DRL network of offline-DHP in our online-DHP approach, since the online-DHP approach is based on the offline DRL network. Figure 13 shows the MO results of our online-DHP approach with and without the offline DRL network. As observed in this figure, the offline DRL network is able to increase MO results of our online-DHP approach, for all 15 sequences. In addition, the MO value can be increased from 0.502 to 0.753 on average, when the offline DRL network is incorporated in online-DHP. Therefore, the learned DRL network of offline-DHP also benefits the online prediction of HM positions in online-DHP.

In online-DHP, the performance of online HM prediction is related to the hyperparameters of th_{MO} , XXX, and XXX. In the following, we analyze the influences of this hyperparameters on the performance of online-DHP. Figure XXX shows the MO results at various values of th_{MO} , XXX, and XXX. The hyperparameter th_{MO} is the threshold of the MO value achieved by XXX, and it determines whether the *action* of online-DHP is output as the HM scan-path at the current frame. We can see from Figure XXX that XXX is increased at $th_{MO} \in XXX$ and converged once $th_{MO} \geq ???$. Thus, in this paper, we set th_{MO} to be XXX.

Now, we move to the evaluation of Online-DHP. We compare our Online-DHP approach with Deep 360 Pilot approach [18] and two baselines.

Table ?? compares mean MO results of our and other approaches, in predicting HM positions over all 15 test sequences of panoramic video. Here, MO results are averaged over all frames for each test sequence. The Random-baseline means predicting the HM position in a totally random way. And the Keep-baseline means predicting the HM position due to the HM position of the former frame. We can see from this table that our approach performs far better than other approaches. Specifically, our approach has ? average increment over Deep 360 Pilot approach [18]. We think that is mostly because that our approach has priori knowledge from the Offline-DHP. For the two baselines, the Keep-baseline is a little better than the Random-baseline mostly because of the guidance of the former frame. Compared with the two baselines, our approach is also obviously much more effective.

Offline-DHP is an important component of our approach. Figure 13 shows the result of mean MO between Online-DHP and Online-DHP without Offline-DHP. We can see that the Offline-DHP enhances the performance of online prediction a lot. We think it's mostly because the Offline-DHP provides priori knowledge for the online prediction. And this is very effective in predicting.

After the comparison, we move to the analysis of an important parameter th_{MO} in algorithm ??, which affects both the result of MO(6) and the time complexity of our approach. We call it the threshold of MO. According to our approach, we think the threshold th_{MO} is positively correlated with the result of MO, but negatively correlated with the time complexity.

7 CONCLUSION

In this paper, we have proposed the DHP approach to predict HM positions on panoramic video. First, we established a new database named PVS-HM, which includes HM data of 58 subjects in 76

panoramic sequences. We found for our database that the HM positions are highly consistent across humans. Thus, the consistent HM positions on each panoramic frame can be represented in the form of the HM map, which encodes possibility of each pixel being HM position. Second, we proposed the offline-DHP approach to estimate HM maps in an offline manner. To be more specific, our offline-DHP approach leverages DRL to make decision on *actions* of HM scan-paths. Afterwards, HM scan-paths of several *agents* from multiple DRL workflows are integrated to yield final HM maps. Third, we proposed the online-DHP approach, which online predicts the HM positions of one subject. In online-DHP, the DRL algorithm was developed in determining the HM positions of one *agent* at the incoming frames, given *observation* of the previous HM scan-path and current video content. The DRL algorithm is based on the learned model of offline-DHP in extracting spatio-temporal features of attention-related content. Finally, the experimental results showed that offline-DHP and online-DHP are superior to other conventional approaches, in the offline and online tasks of HM prediction for panoramic video.

Humans always perceive the world around them in a panorama, rather than the 2D plane. Therefore, modelling attention on panoramic video is an important component in establishing human-like computer vision systems. Our work at the current stage mainly focuses on predicting HM positions, as the first step towards attention modelling of panoramic video. The future work should further predict eye fixations within FoV regions of panoramic video. The potential applications of our approach in human-like computer vision systems are another promising work in future. For example, the online-DHP approach may be embedded in robotics, to mimic human's way in perceiving the real world. Besides, panoramic video has large perceptual redundancy, since most of panoramic regions cannot be seen by humans. It is thus possible to use the offline-DHP approach to remove such perceptual redundancy, for saving bit-rates of panoramic video coding.

APPENDIX A PROOF OF PROPOSITION 1

Example.

A.1 The Analysis of FCB Combined in HM Maps

The literature of saliency detection [49] has argued that human attention has strong center bias in images or videos, and that the incorporation of center bias can improve the performance of saliency detection. Similarly, there exists FCB in viewing panoramic videos, as discussed in *Finding 2*. Hence, this appendix discusses the combination of the offline-DHP approach and the FCB feature. Here, we apply the FCB feature as an additional channel in generating HM maps of panoramic videos. Specifically, assume that \mathbf{H}^f is the HM map of the FCB feature. Similar to the center bias feature of image saliency detection [49], we apply the following 2D Gaussian distribution to model \mathbf{H}^f for each frame,

$$\mathbf{H}^f(u, v) = \exp\left(-\frac{(u - u_f)^2 + (v - v_f)^2}{\sigma_f^2}\right), \quad (7)$$

where (u, v) is the longitude and latitude of the GDS location in the map, and (u_f, v_f) is the longitude and latitude of the front center position in GDS. In addition, σ_f is the standard deviation of the 2D Gaussian distribution.

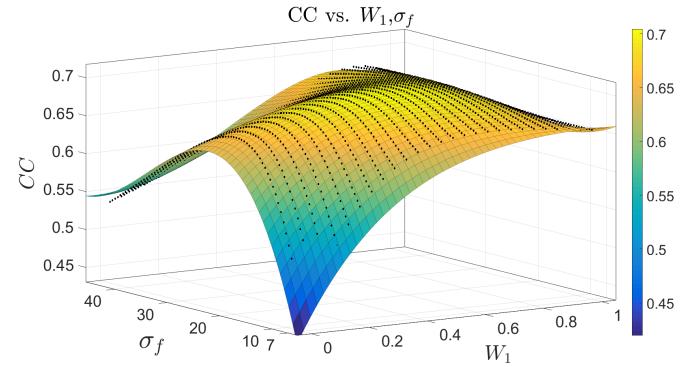


Fig. 14: The fitting surface of CC between the predicted and ground-truth HM maps at various σ_f and w_1 . The dark dots in this figure represent the CC results at each specific value of σ_f and w_1 , which are used to fit the surface. Note that the CC results are obtained over all training data of the PVS-HM database

Next, we need to combine \mathbf{H}^f with the predicted HM map \mathbf{H}_t for each panoramic frame by

$$\mathbf{H}_t^c = w_1 \cdot \mathbf{H}^f + w_2 \cdot \mathbf{H}_t, \quad (8)$$

where \mathbf{H}_t^c is the HM map integrated with the FCB feature for frame t . In (8), w_1 and w_2 are the weights corresponding to the channels of \mathbf{H}_c and \mathbf{H}_t , respectively. Given (7) and (8), the following optimization formulation is applied to obtain the values of σ_f , w_1 and w_2 :

$$\min_{w_1, w_2} \sum_{t=1}^T \text{CC}(\mathbf{H}_t^c, \mathbf{H}_t^g), \quad \text{s.t. } w_1 + w_2 = 1. \quad (9)$$

In (8), \mathbf{H}_t^g is the ground-truth HM map of each frame; $\text{CC}(\cdot, \cdot)$ indicates the CC value of two maps. Then, we solve the above optimization formulation by the least square fitting over all training data of our PVS-HM database. Consequently, the optimal values of σ_f , w_1 and w_2 are 21.1° , 0.52 and 0.48 , respectively. These values are used in integrating the FCB feature in our offline-FCB approach. Note that the same way is applied to obtain weights of w_1 and w_2 when combining the FCB feature with other approaches.

Figure 14 shows the results of CC between the predicted and ground-truth HM maps at various values of σ_f and w_1 . From this figure, we can see that CC varies from 0.44 to 0.70 alongside the increase of w_1 from 0 to 1, reaching the maximum value at $\sigma_f = 21.1^\circ$ and $w_1 = 0.52$. This indicates that both the FCB feature and our offline-DHP approach are effective in predicting the HM maps of panoramic video, and that the effectiveness of the FCB feature is different at varying combination weights. In addition, as shown in Figure 14, at $w_1 = 0.52$, the CC value increases from 0.66 to 0.70, when σ_f grows from 7° to 21.1° , and then it decreases to 0.63 till $\sigma_f = 43.6^\circ$. Thus, the standard deviation of the 2D Gaussian distribution is set to be 21.1° for the FCB feature in our experiments.

ACKNOWLEDGMENTS

The authors would like to thank...

REFERENCES

- [1] U. Neumann, T. Pintaric, and A. Rizzo, "Immersive panoramic video," in *ACM MM*, 2000.

- [2] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE TPAMI*, vol. 35, no. 1, pp. 185–207, Jan. 2013.
- [3] Y. S. de la Fuente, R. Skupin, and T. Schierl, "Video processing for panoramic streaming using hevc and its scalable extensions," *Multimedia Tools and Applications*, pp. 1–29, 2016.
- [4] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE TMM*, vol. 18, no. 9, pp. 1819–1831, 2016.
- [5] M. Xu, C. Li, Y. Liu, X. Deng, and J. Lu, "A subjective visual quality assessment method of panoramic videos," in *ICME*, July 2017, pp. 517–522.
- [6] M. Stengel and M. Magnor, "Gaze-contingent computational displays: Boosting perceptual fidelity," *IEEE SPM*, vol. 33, no. 5, pp. 139–148, 2016.
- [7] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971–1984, 2008.
- [8] Y.-C. Su, D. Jayaraman, and K. Grauman, "Pano2vid: Automatic cinematography for watching 360-degree videos," in *ACCV*, 2016.
- [9] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *ICML*, 2016.
- [10] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [11] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE TIP*, vol. 13, no. 10, pp. 1304–1318, 2004.
- [12] Y. Lin, Y. Y. Tang, B. Fang, Z. Shang, Y. Huang, and S. Wang, "A visual-attention model using earth mover's distance-based saliency measurement and nonlinear feature combination," *IEEE TPAMI*, vol. 35, no. 2, pp. 314–328, Feb. 2013.
- [13] J. Zhang and S. Sclaroff, "Exploiting surroundedness for saliency detection: a boolean map approach," *IEEE TPAMI*, vol. 38, no. 5, pp. 889–902, 2016.
- [14] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proceedings of ICCV*, 2015, pp. 262–270.
- [15] Y. Liu, S. Zhang, M. Xu, and X. He, "Predicting salient face in multiple-face videos," in *CVPR*, 2017.
- [16] T. Löwe, M. Stengel, E.-C. Förster, S. Grogorick, and M. Magnor, "Visualization and analysis of head movement and gaze data for immersive video in head-mounted displays," in *ETVIS*, 2015.
- [17] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [18] 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 360 sports video," in *CVPR*, 2017.
- [19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [20] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vision research*, vol. 49, no. 10, pp. 1295–1306, Jun. 2009.
- [21] G. Bocciognone, "Nonparametric bayesian attentive video analysis," in *International Conference on Pattern Recognition (ICPR)*, 2008.
- [22] L. Zhang, M. H. Tong, and G. W. Cottrell, "Sunday: Saliency using natural statistics for dynamic analysis of scenes," in *Annual Cognitive Science Conference*, 2009, pp. 2944–2949.
- [23] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE TIP*, vol. 19, no. 1, pp. 185–198, Jan. 2010.
- [24] Z. Ren, S. Gao, L.-T. Chia, and D. Rajan, "Regularized feature reconstruction for spatio-temporal saliency detection," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3120–3132, Aug. 2013.
- [25] S. Hosseini Khatoonabadi, N. Vasconcelos, I. V. Bajic, and Y. Shan, "How many bits does it take for a stimulus to be salient?" in *CVPR*, 2015.
- [26] M. Xu, L. Jiang, X. Sun, Z. Ye, and Z. Wang, "Learning to detect video saliency with hevc features," *IEEE TIP*, vol. 26, no. 1, pp. 369–385, 2017.
- [27] S. S. Kruthiventi, K. Ayush, and R. V. Babu, "Deepfix: A fully convolutional neural network for predicting human eye fixations," *arXiv preprint arXiv:1510.02927*, 2015.
- [28] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *European Conference on Computer Vision*. Springer, 2016, Conference Proceedings, pp. 825–841.
- [29] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," *arXiv*, 2016.
- [30] C. Bak, A. Erdem, and E. Erdem, "Two-stream convolutional networks for dynamic saliency prediction," *arXiv*, 2016.
- [31] W. Wang, J. Shen, and L. Shao, "Deep learning for video saliency detection," *arXiv preprint arXiv:1702.00871*, 2017.
- [32] S. Minut and S. Mahadevan, "A reinforcement learning model of selective visual attention," in *AAMAS*, 2001.
- [33] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *NIPS*, 2014.
- [34] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," *arXiv*, 2016.
- [35] Z. Wang, T. Schaul, M. Hessel, H. v. Hasselt, M. Lanctot, and N. d. Freitas, "Dueling network architectures for deep reinforcement learning," in *ICML*, 2016.
- [36] J. Foote and D. Kimber, "Flycam: Practical panoramic video and automatic camera control," in *ICME*. IEEE, 2000.
- [37] X. Sun, J. Foote, D. Kimber, and B. Manjunath, "Region of interest extraction and virtual camera control based on panoramic video capturing," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 981–990, 2005.
- [38] Y.-C. Lin, Y.-J. Chang, H.-N. Hu, H.-T. Cheng, C.-W. Huang, and M. Sun, "Tell me where to look: Investigating ways for assisting focus in 360 video," in *ACM CHI*, 2017.
- [39] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE TCSVT*, vol. 22, no. 12, pp. 1649–1668, 2013.
- [40] J. Li, C. Xia, Y. Song, S. Fang, and X. Chen, "A data-driven metric for comprehensive evaluation of saliency models," in *ICCV*, 2015.
- [41] E. Matin, "Saccadic suppression: a review and an analysis," *Psychological bulletin*, vol. 81, no. 12, p. 899, 1974.
- [42] M. J. Frederic P. Miller, Agnes F. Vandome, "Mean of circular quantities," *Mathematics*, 2010.
- [43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2014.
- [44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [45] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *AAAI*, 2015.
- [46] B. Shumaker and R. Sinnott, "Virtues of the haversine," *Sky and telescope*, vol. 68, pp. 158–159, 1984.
- [47] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3–4, pp. 279–292, 1992.
- [48] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.
- [49] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Computer Vision, 2009 IEEE 12th international conference on*, 2009.

Michael Shell Biography text here.

PLACE
PHOTO
HERE

John Doe Biography text here.

Jane Doe Biography text here.