

# 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 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 an agent. Based on our findings, we propose a DRL based HM prediction (DHP) approach in 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 at the previous frames. Such online estimation is achieved by developing a DRL algorithm, which is 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 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 state-of-the-art saliency detection methods [2]. 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], rendering [5], synopsis [6], and automatic cinematography [7].

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) [8], this paper proposes a DRL based HM prediction (DHP) approach for modelling attention on panoramic video. 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 means 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.* [9], in which the features of color, intensity and orientation are combined to generate the saliency map of an image. Later, Itti *et al.* [10] proposed to add two features in [9], 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 [11] and the boolean map based saliency model (BMS) [12]. Most recently, deep learning has been successfully applied in video saliency detection, such as SALICON [13] and Liu's approach [14]. 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 [15]. 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 represented by the longitude and latitude in the geographic coordinate system (GDS) [16]. 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.*

*Manuscript received April 19, 2005; revised August 26, 2015.*



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 filter<sup>1</sup>.

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 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 *actions* of multiple *agents* and takes viewed panoramic content as the *observation of environment*. As such, the DRL model of HM prediction can be learned for HM prediction. In training the DRL model, *reward* is designed to measure the difference of *actions* of 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, HM maps of panoramic video are generated upon HM positions, obtained from scan-paths of several *agents* in multiple DRL workflows.

For online HM prediction, the latest work of [17] proposed a deep 360 pilot, which automatically shifts viewing direction (equivalent to 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) [18] 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 [17]. 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, in terms of longitude and latitude. Different from [17], our online-DHP approach does not need to detect the salient object of 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 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, with a thorough analysis on their HM data across 76 panoramic video sequences.
- 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 at the current and previous frames.

## 2 RELATED WORK

### 2.1 Saliency detection

The only approach on predicting HM positions of panoramic videos is the most recent work of [7], 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 saliency maps of video frames. The studies on visual saliency start from images in 1998, when Itti and Koch [9] 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 [10], in which two dynamic features of motion and flicker contrast are combined with [9] for detecting saliency in 2D videos. Both [9] and [10] can be seen as heuristic approaches for saliency detection, since they make use of the understanding of the HVS to develop the computational models. Recently, some advanced heuristic approaches, e.g., [11], [12], [19], [20], [21], [22], [23],

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

[24], [25], have been proposed for saliency detection in 2D videos. Specifically, [19] 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 [19] for video saliency detection. Some other Bayesian frameworks [20], [21] were also developed for detecting video saliency. Besides, Lin *et al.* [11] 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.* [12] explored the surround cue for saliency detection, by characterizing a set of binary images with randomly thresholds on color channels. Recently, [24] and [25] have investigated that some features (e.g., motion vector) in compressed domain are of high correlation with human attention, thus being explored in video saliency detection.

Benefiting from the most recent success of deep learning, deep neural networks (DNNs) [13], [14], [26], [27], [28], [29], [30] 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, [28] 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.* [14] combined CNN and multi-stream LSTM for detecting saliency in videos with multiple faces. Besides, other DNN structures have been developed to detect either static saliency [26], [27] or dynamic saliency [28], [29], [30].

Although saliency detection has been thoroughly studied for predicting eye movement on 2D videos, there is no work on the prediction of 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, for modelling attention on panoramic videos. Towards the HM maps of panoramic videos, a DRL approach is developed to estimate the *actions* of HM by multiple *agents* 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., [31], [32], [33], [34].

## 2.2 Virtual cinematography

Virtual cinematography of panoramic videos was proposed in [7], [17], [35], [36], [37], which directs an imaginary camera to virtually capture natural FOV (NFOV). In general, virtual cinematography attempts to agree with HM positions of humans at each panoramic video frame. The early work of [35] proposed cropping object-of-interest in panoramic videos, such that NFOV can be generated for virtual cinematography. Later, in [36] the cropped object-of-interest is tracked across frames by a Kalman filter, for automatically controlling virtual camera in virtual cinematography of panoramic videos. The approach of [36] can work on both compressed

and uncompressed domains, as two methods were developed for detecting object-of-interest in compressed and uncompressed domains, respectively. Both the works of [35], [36] 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 humans.

Most recently, data-driven approaches boost the development of virtual cinematography for panoramic videos. Specifically, Pano2Vid [7] 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 has been studied [17], [37] in a data-driven way, in which HM positions are predicted by an online mechanism. Specifically, the deep 360 pilot was proposed in [17], 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 position at previous frames. Consequently, the 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 videos 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* [37] investigated two focus assistance techniques to help the subject track the key object in viewing panoramic videos, 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 [7], [17], [35], [36], [37] all depend on the detector of object-of-interest. Thus, they can be only applied for some specific panoramic videos with salient objects, such as video conferencing or classroom scenes in [35], [36] and the sports videos in [7], [17], [37]. Different from these conventional approaches, our online HM prediction 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.

## 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 PVS-HM database. Our PVS-HM database allows quantitative analysis of human's 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 VRCun, with resolution ranging from 3K to 8K. As seen in Table 1, the content of these sequences are 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 [38] without any change at bit-rates. Note that the audio tracks were



Fig. 2: 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. 3: 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.

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 viewing 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, 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 five 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  $H_A$  and  $H_B$ , respectively. For a panoramic frame, we quantify the correlation of the HM maps between  $H_A$  and  $H_B$  using linear correlation coefficient (CC) [39]. Table 1 lists the averaged CC ( $\pm$  standard deviations) of HM maps between Groups  $A$  and  $B$ , over all frames for each sequence. We further show in Table 1 the CC values between  $H_A$  and the maps with Gaussian distribution, uniform distribution and FCB. It can be seen from this table that the CC values are rather high across different

sequences. It can be also seen from this table that the average CC value between  $H_A$  and  $H_B$  over all 76 panoramic sequences is XXX, with the standard deviation being XXX. This CC value is significantly larger than that between  $H_A$  and random maps or FCB maps. Thus, it is obvious that HM positions of subjects are highly consistent. This completes the analysis of *Finding 1*.

*Finding 2: The magnitude of HM scan-paths is similar across subjects in viewing the same panoramic video.*

*Analysis:* Human 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 2 plots the mean and standard deviation of HM scan-path magnitude across all 58 subjects. The last column of Figure 2 further shows the mean and standard deviation, 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 the averaged results. Thus, we can conclude that there exists similarity for the magnitude of HM scan-paths across subjects, when the subjects view the same panoramic sequence. This completes the analysis of *Finding 2*.

*Finding 3: 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 consistent HM regions (i.e., regions with similar HM positions), 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 has HM positions of at least 12 subjects within a small *great-circle distance* range of  $3^\circ$  [40]. For each panoramic video sequence, Figure 3 shows the circular standard deviation [41] of HM scan-path directions, starting from the consistent HM regions within one-

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	AcerPredator	0.839±0.087	Driving	AirShow	0.783±0.078	Others	A380	0.839±0.106	Video Game	CS	0.819±0.084
	BFG	0.644±0.146		DrivingInAlps	0.857±0.071		CandyCarnival	0.723±0.094		Dota2	0.714±0.103
	CMLauncher	0.828±0.119		F5Fighter	0.592±0.126		MercedesBenz	0.592±0.133		GalaxyOnFire	0.762±0.084
	Cryogenian	0.526±0.174		HondaF1	0.872±0.053		RingMan	0.897±0.054		LOL	0.724±0.097
	LoopUniverse	0.779±0.078		Rally	0.867±0.047		RioOlympics	0.624±0.123		MC	0.726±0.115
	Pokemon	0.607±0.182		Supercar	0.854±0.064		VRBasketball	0.770±0.105		SuperMario64	0.860±0.054
Movie	Help	0.859±0.122	Scenery	Antarctic	0.674±0.135	Show	BTSRun	0.867±0.061	Action Sports	Gliding	0.528±0.158
	IRobot	0.771±0.078		BlueWorld	0.559±0.156		Graffiti	0.807±0.100		Parachuting	0.628±0.157
	Predator	0.696±0.124		Dubai	0.646±0.133		KasabianLive	0.722±0.132		RollerCoaster	0.834±0.078
	ProjectSoul	0.918±0.053		Egypt	0.665±0.131		NotBeAloneTonight	0.587±0.131		Skiing	0.766±0.104
	StarWars	0.950±0.016		StarryPolar	0.495±0.152		Symphony	0.779±0.096		Surfing	0.830±0.096
	Terminator	0.843±0.078		WesternSichuan	0.667±0.138		VRBasketball	0.770±0.105		Waterskiing	0.781±0.128
	Overall						0.745±0.114				

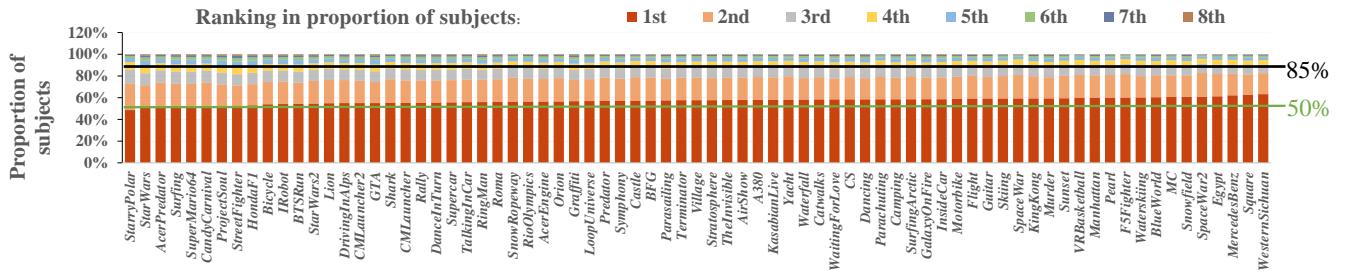


Fig. 4: Distribution of HM scan-path over 8 discrete directions, for all 76 sequences from the PVS-HM database. Note that the proportions of subjects, whose HM scan-path directions fall into each discrete direction, are ranked and then shown in this figure.

second time slot<sup>2</sup>. The last column of Figure 3 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  $38.1^\circ$ , and it is significantly less than  $103.9^\circ$  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 3* can be validated.

*Finding 4:* Almost 50% subjects are consistent with one HM scan-path direction (among 8 uniform-quantized directions), and over 85% subjects are consistent with 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 3*) 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 4. 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. We further show in Figure 5 the proportions of subjects averaged over all 76 sequences, the HM scan-path directions of which rank 1-st to 8-th. As seen in this

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

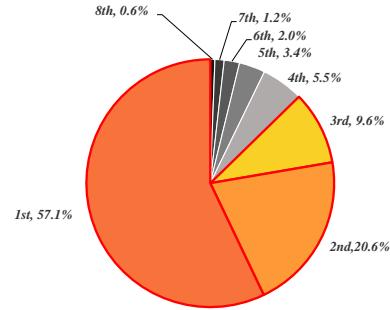


Fig. 5: Proportions of subjects with scan-path falling into each of 8 ranked directions. The proportions are averaged over all 76 sequences in the PVS-HM database.

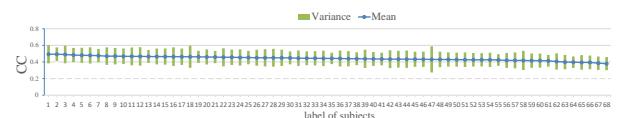


Fig. 6: Correlation coefficient of HM scan-paths over different time intervals and its variance.

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 4* can be validated.

*Finding 5:* For one subject, the HM scan-path at the current time interval is correlated with that at the previous time interval and observed video content.

*Analysis:* For each individual subject, the vectors of HM scan-

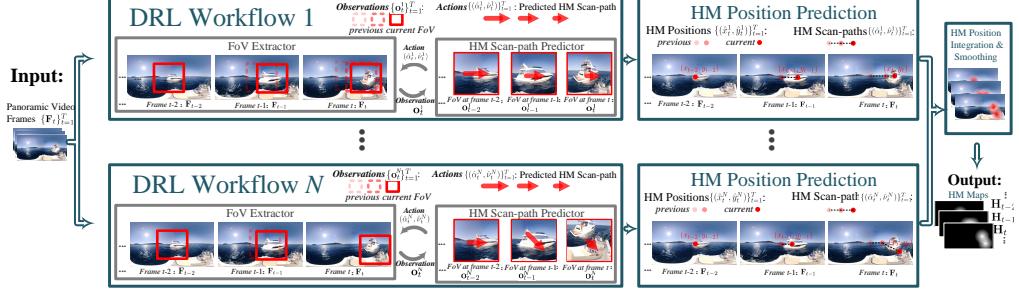


Fig. 7: Framework of our Offline-DHP approach.

TABLE 2: Notations denoted in Figure 7

$\{t\}_1^T$	The frames of panoramic video from 1 to $T$ , with $t$ being the currently processed frame
$\{\mathbf{F}_t\}_{t=1}^T$	The panoramic frames from 1 to $T$ , as the input to offline-DHP/online-DHP
$\{\mathbf{H}_t\}_{t=1}^T$	The HM maps of 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/workflow
$\{n\}_{n=1}^N$	The DRL workflows from 1 to $N$ , with $n$ being the $n$ -th workflow
$\{\mathbf{o}_t^n\}_{t=1}^T$	The FoV from 1 to $T$ , as the <i>observation</i> of the $n$ -th DRL workflow
$(x_t^m, y_t^m)$	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 workflow at frame $t$ , with one workflow modelling a subject
$\pi_t$	The predicted probability distribution of HM direction at frame $t$ , as the <i>policy</i> <sup>3</sup> of the DRL agent
$\hat{o}_t^n$	The predicted HM direction at frame $t$ from the $n$ -th DRL workflow, as the <i>action</i> of the DRL agent
$\alpha_t^m$	The ground-truth HM direction of the $m$ -th subject at frame $t$
$\hat{v}_t^n$	The predicted HM magnitude at frame $t$ from the $n$ -th DRL workflow, as the <i>action</i> of the DRL agent
$v_t^m$	The ground-truth HM magnitude of the $m$ -th subject at frame $t$
$r_{n,t}^{\alpha}$	The estimated reward for deciding $\hat{o}_t^n$ , as the <i>reward</i> of the $n$ -th DRL workflow
$r_{n,t}^{\nu}$	The estimated reward for deciding $\hat{v}_t^n$ , as the <i>reward</i> of the $n$ -th DRL workflow
$\mathbf{f}_t^n$	The extracted LSTM feature at frame $t$ , as part of <i>observed state</i> of the $n$ -th DRL workflow

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 correlation coefficient (CC) of HM scan-path vectors between two successive time periods, and then average the CC value over all 76 sequences for each subject. Figure 6 shows 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, [17] shows that the current HM scan-path of one subject is also related to observed video content. The above completes the validation of *Finding 5*.

## 4 OFFLINE-DHP APPROACH

### 4.1 Framework of offline-DHP

In the following, we present our offline-DHP approach, in light of our findings in Section 3.2. Figure 7 shows the overall framework of our approach, in which multiple DRL workflows are embedded to generate HM maps of the input panoramic video frames. The notations used in this figure are listed in Table 2.

As shown in Figure 7, 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 panoramic video, seen as the output of our online-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

video,  $\mathbf{H}_t$  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 4* 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  $N$  at each frame is equivalent to  $M$ , for predicting HM positions of all subjects. Next, 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 all 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  $\pi_t$ .

In fact,  $(\hat{x}_t^n, \hat{y}_t^n)$  of one DRL workflow can be modeled by determining a series of *actions*:  $\hat{o}_1^n, \hat{o}_2^n, \dots, \hat{o}_t^n$  and  $\hat{v}_1^n, \hat{v}_2^n, \dots, \hat{v}_t^n$ . It is worth pointing out that it is reasonable to model  $\{\hat{o}_t^n\}_{t=1}^T$  and  $\{\hat{v}_t^n\}_{t=1}^T$  as the *actions* in the DRL workflow, since *Findings 2* and *3* have indicated that subjects are consistent in the magnitudes and directions of HM scan-paths. As can be seen in Figure 7, in a single DRL 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 7 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  $(\hat{o}_{t-1}^n, \hat{v}_{t-1}^n)$

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 in our offline-DHP approach. In this section, we take the  $n$ -th workflow as an example. Figure 8 shows the proposed training model of DRL. As shown in Figure 8, 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* is estimated to evaluate the *action* of the DRL model, which measures the similarity between the prediction and ground-truth of HM scan-paths. 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 by the following procedure.

- (1) At frame  $t$ , the FoV extractor obtains the current *observation*  $\mathbf{o}_t^n$  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 the content of FoV ( $103^\circ \times 60^\circ$ ) in the  $n$ -th workflow. It is projected to 2D region and then down-sampled to  $42 \times 42$ .
- (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 [42] and one LSTM layer [43], [44], which are used to extract spatial and temporal features, respectively. The details about the architecture of the DRL network can be found in Figure 8.
- (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  $\pi_t$ . Here,  $\pi_t$  is the probability distribution over the *actions* of HM scan-path directions.
- (4) Given  $\pi_t$ , the HM scan-path predictor randomly samples an *action*  $\hat{\alpha}_t^n$ , with standard deviation  $\varepsilon$  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  $(\hat{x}_{t+1}^n, \hat{y}_{t+1}^n)$ , and the *reward* estimator returns the *rewards*  $r_{n,t}^\alpha$  and  $r_{n,t}^\nu$  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  $t$ .
- (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 *reward*:  $r_{n,t}^\alpha$  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 *reward*  $r_{n,t}^\alpha$  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  $\{(x_t^n, y_t^n)\}_{m=1}^M$  in the *rewards* of  $r_{n,t}^\alpha$  and  $r_{n,t}^\nu$ , and the 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 *phase difference*, and  $D_s$  denotes *great-circle distance* [45]. Moreover,  $\rho$  and  $\varrho$  are the standard deviations of Gaussian distributions, as the hyper-parameters. Similarly, we have

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

where  $\varsigma$  is the hyper-parameter for the standard deviation of HM scan-path magnitude.

**Optimization.** Now, we need to optimize *reward*:  $r_t^\alpha$  and  $r_t^\nu$ , when learning the network parameters of our DRL model in Figure 8. Our offline-DHP approach implements the asynchronous DRL method [8]. Hence, multiple workflows are run to interact with multiple *environments* with workflow-specific parameter vectors  $\{\theta_\nu, \theta_\pi, \theta_V\}$ , producing  $\hat{\nu}_t^n$ ,  $\hat{\pi}_t^n$  and  $V$ . Here,  $V$  denotes *state value* output by the DRL network, which is obtained using the same way as [8]. Meanwhile, global-shared parameter vectors  $\{\theta_\nu, \theta_\pi, \theta_V\}$ <sup>4</sup> are updated via accumulating gradient. For more details about the workflow-specific and global-shared parameter vectors, refer to [8]. In our approach, *reward*  $r_t^\nu$  is optimized to train  $\theta_\nu$  as follows,

$$d\theta_\nu \leftarrow d\theta_\nu + \nabla_{\theta'_\nu} \sum_{t=1}^T r_t^\nu. \quad (3)$$

Besides, we can optimize *reward*  $r_t^\alpha$  by

$$d\theta_V \leftarrow d\theta_V + \nabla_{\theta'_V} \sum_{t=1}^T \left( \sum_{i=t}^T \gamma^{i-t} r_i^\alpha - V(\mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta'_V) \right)^2, \quad (4)$$

$$\begin{aligned} d\theta_\pi \leftarrow d\theta_\pi + \nabla_{\theta'_\pi} \sum_{t=1}^T & \log \pi(\hat{\alpha}_t^n | \mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta'_\pi) \cdot \\ & \left( \sum_{i=t}^T \gamma^{i-t} r_i^\alpha - V(\mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta'_V) \right), \end{aligned} \quad (5)$$

where  $\gamma$  is the discount factor of *Q-learning* [46]. In addition,  $V(\mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta'_V)$  denotes state value  $V$  yielded by  $\mathbf{o}_t^n$ ,  $\mathbf{f}_{t-1}^n$  and  $\theta'_V$ ;  $\pi(\hat{\alpha}_t^n | \mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta'_\pi)$  stands for the probability of *action*  $\hat{\alpha}_t^n$  that is yielded by policy  $\pi_t$  from  $\mathbf{o}_t^n$ ,  $\mathbf{f}_{t-1}^n$  and  $\theta'_\pi$ . Finally, upon the above equations, RMSProp [47] is applied to optimize *reward* 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. Our online-DHP approach refers to predicting the  $n$ -th subject's HM position  $(\hat{x}_{t+1}^n, \hat{y}_{t+1}^n)$  at the next frame, given his/her HM positions

4. As can be seen in Figure 8,  $\{\theta_\nu, \theta_\pi, \theta_V\}$  share all CNN and LSTM layers in our offline-DHP approach, but they are separated at the output layer.

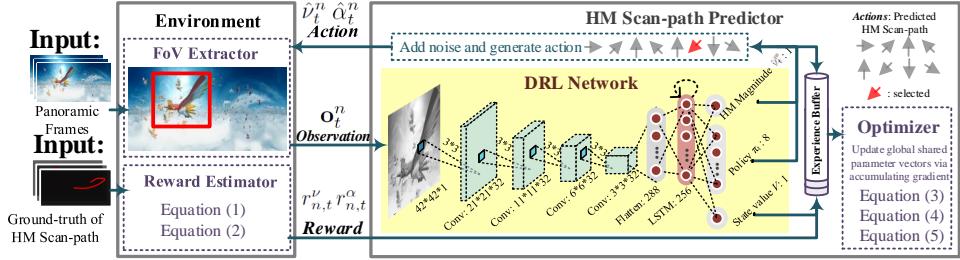


Fig. 8: DRL model for the HM scan-path predictor.

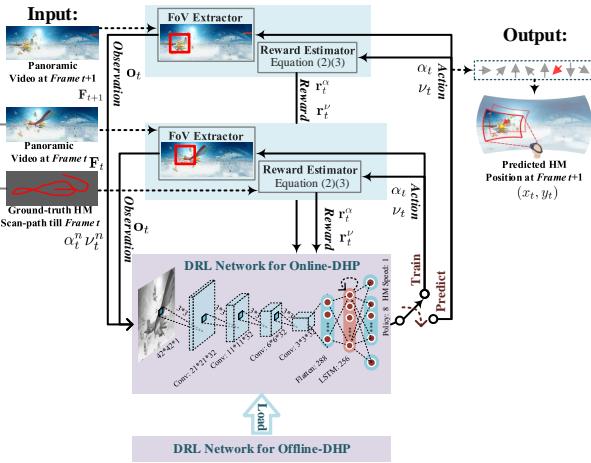


Fig. 9: Framework of our Online-DHP approach.

$\{(x_1^n, y_1^n), \dots, (x_t^n, y_t^n)\}$  till the current frame. Here, we define this subject as a *viewer*. Figure 9 shows the framework of our online-DHP approach. As shown in this framework, the input to our online-DHP approach is the viewer's HM scan-path  $\{(\alpha_1^n, \nu_1^n), \dots, (\alpha_{t-1}^n, \nu_{t-1}^n)\}$  and frame content  $\{F_1, \dots, F_t\}$  till frame  $t$ , while the output is the predicted HM position  $(\hat{x}_{t+1}^n, \hat{y}_{t+1}^n)$  at frame  $t+1$  for the *viewer*. This can be seen as online prediction of HM positions  $\{(\hat{x}_t^n, \hat{y}_t^n)\}_{t=1}^T$ . 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 in more details.

### 5.1 Stage I: Training

At the beginning, the HM position  $(\hat{x}_1^n, \hat{y}_1^n)$  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. 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  $(\alpha_i^n, \nu_i^n)$  are the predicted and ground-truth *action* of the HM scan-path, and  $\mathbf{o}_i^n$  is the *observation* of the FoV content for the subject.
- 2) Take *action* of  $(\hat{\alpha}_i, \hat{\nu}_i)$  upon the current *observation*  $\{\mathbf{o}_1^n, \dots, \mathbf{o}_i^n\}$  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 *reward*  $(r_{n,i}^{\nu}, r_{n,i}^{\nu})$  from the *reward estimator* with (1) and (2), which measures how close the *action*  $(\hat{\alpha}_i^n, \hat{\nu}_i^n)$  to the ground-truth HM scan-path  $(\alpha_i^n, \nu_i^n)$  of the  $n$ -th *viewer*. Here, the ground-truth HM scan-path used in (1) and (2) is from the single *viewer*, rather than from all subjects.
- 4) Generate new *observation*  $\mathbf{o}_{i+1}^n$  from the FoV extractor with above *action*  $(\hat{\alpha}_i^n, \hat{\nu}_i^n)$ , and then it is input to the DRL network.
- 5) Update the DRL network using (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 reinforce learning as that of one DRL workflow of offline-DHP, which is 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 termination conditions. In our approach, there are two criteria for the termination condition. The first criterion is the maximal number  $E$  of episodes, which is set to be 500 in this paper. The second one is based on mean overlap (MO), which measures how close the predicted HM position to the ground-truth HM position. MO ranges from 0 to 1, where larger MO indicates more precise prediction. Specifically, MO is defined as,

$$MO = \frac{A(FoV_p \cap FoV_g)}{A(FoV_p \cup FoV_g)}, \quad (6)$$

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 conditions. Algorithm ?? 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 9 is turned to "predict" and the DRL network make a prediction for action at frame  $t + 1$ . Note that 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 trained model is used to produce the HM position as follows.

With the trained DRL model at the first stage, XXX

The direction and magnitude of HM scan-path, i.e.,  $\hat{\alpha}_t$  and  $\hat{\nu}_t$ , can be obtained at frame  $t + 1$ , as the *action* of DRL. As a result, the HM position  $(\hat{x}_{t+1}^n, \hat{y}_{t+1}^n)$  at frame  $t + 1$  can be predicted, given the ground-truth HM position  $(x_t^n, y_t^n)$  at frame  $t$ . Algorithm XXX presents the summary of the prediction stage in online-DHP. Finally, online-DHP is achieved by training and predicting till the last frame.

---

**Algorithm 1** The algorithm of online-DHP for subject  $m$  when predicting  $t + 1$ : Stage Training.

---

**Input:** Panoramic video and viewer's ground-truth HM position  $\{(x_i^m, y_i^m)\}_{i=1}^t$ ;  
 Load offline-DHP model to initialize model with parameter vectors:  $\{\theta_{\hat{\nu}}, \theta_{\hat{\pi}}, \theta_V\}$ ;  
**for**  $e = 1$  **to**  $E$  **do**  
   Initialize HM position:  $\hat{x}_1 = 0, \hat{y}_1 = 0$ ;  
   Initialize LSTM feature:  $\mathbf{f}_0 = \mathbf{0}$ ;  
   **for**  $i = 1$  **to**  $t - 1$  **do**  
     Extract FoV  $\mathbf{o}_i$  according to  $(\hat{x}_i, \hat{y}_i)$ ;  
     Obtain policy and LSTM feature:  $\hat{\pi}_i, \mathbf{f}_i = \hat{\pi}(\mathbf{o}_i, \mathbf{f}_{i-1}; \theta_{\hat{\pi}})$ ;  
     Select  $\hat{\alpha}_i$  according to the  $\epsilon$ -greedy policy based on  $\hat{\pi}_i$ ;  
     Obtain HM magnitude:  $\hat{\nu}_i = \hat{\nu}(\mathbf{o}_i, \mathbf{f}_{i-1}; \theta_{\hat{\nu}})$ ;  
     According to  $\hat{\alpha}_i, \hat{\nu}_i$ , update  $(\hat{x}_i, \hat{y}_i)$  to  $(\hat{x}_{i+1}, \hat{y}_{i+1})$ ;  
     Estimate reward  $r_i^\nu, r_i^\alpha$  of  $\hat{\alpha}_i, \hat{\nu}_i$  according to (1), (2);  
     Compute MO <sub>$i$</sub>  of  $(\hat{x}_i, \hat{y}_i)$  and  $(x_i^m, y_i^m)$ ;  
     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  $\{\theta_{\hat{\nu}}, \theta_{\hat{\pi}}, \theta_V\}$  according to (3), (4), (5) but replace  $\{\theta'_{\hat{\nu}}, \theta'_{\hat{\pi}}, \theta'_V\}$  with  $\{\theta_{\hat{\nu}}, \theta_{\hat{\pi}}, \theta_V\}$ ;  
    $e \leftarrow e + 1$ ;  
    $MO = \frac{\sum_i MO_i}{t-1}$ ;  
   **if**  $MO > th_{MO}$  **then**  
     **break**;  
   **end if**  
**end for**  
**Return:** Trained parameter vectors:  $\{\theta_{\hat{\nu}}, \theta_{\hat{\pi}}, \theta_V\}$ ;

---

## 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

---

**Algorithm 2** The algorithm of online-DHP for subject  $m$  when predicting  $t + 1$ : Stage Predicting.

---

**Input:** Trained parameter vectors:  $\{\theta_{\hat{\nu}}, \theta_{\hat{\pi}}, \theta_V\}$  from stage training;  
 Load trained parameter vectors from stage training:  $\{\theta_{\hat{\nu}}, \theta_{\hat{\pi}}, \theta_V\}$ ;  
 Initialize LSTM feature:  $\mathbf{f}_0 = \mathbf{f}$ ;  
**for**  $i = 1$  **to**  $t - 1$  **do**  
   Extract FoV  $\mathbf{o}_i$  according to  $(x_i^m, y_i^m)$ ;  
   Obtain LSTM feature:  $\mathbf{f}_i = \hat{\pi}(\mathbf{o}_i, \mathbf{f}_{i-1}; \theta_{\hat{\pi}})$ ;  
    $i \leftarrow i + 1$ ;  
**end for**  
 Extract FoV  $\mathbf{o}_t$  according to  $(x_t^m, y_t^m)$ ;  
 Obtain policy:  $\hat{\pi}_t = \hat{\pi}(\mathbf{o}_t, \mathbf{f}_{t-1}; \theta_{\hat{\pi}})$ ;  
 Choose  $\hat{\alpha}_t$  with the greedy policy based on  $\hat{\pi}_t$ ;  
 Obtain HM magnitude:  $\hat{\nu}_t = \hat{\nu}(\mathbf{o}_t, \mathbf{f}_{t-1}; \theta_{\hat{\nu}})$ ;  
 According to  $\hat{\alpha}_t, \hat{\nu}_t$ , update  $(x_t^m, y_t^m)$  to  $(\hat{x}_{t+1}, \hat{y}_{t+1})$ , which is the prediction of HM position at frame  $t + 1$ ;  
**Return:** The online prediction of the HM position for subject  $m$  at frame  $t + 1$ :  $(\hat{x}_{t+1}, \hat{y}_{t+1})$ ;

---

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  $\rho$  and  $\varrho$  of (1) and (2) were tuned over the test set, to estimate *reward* of HM scan-path prediction. As a result,  $\rho$  and  $\varrho$  were set to be 42 and 0.7. In addition, we followed [8] to set other hyperparameters of DRL. For example, we chose the discount factor  $\gamma$  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 [40], 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 [24], BMS [12] and SALICON [13]. In particular, OBDL [24] and BMS [12] are the latest saliency detection approaches for video and image, respectively. SALICON [13] 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

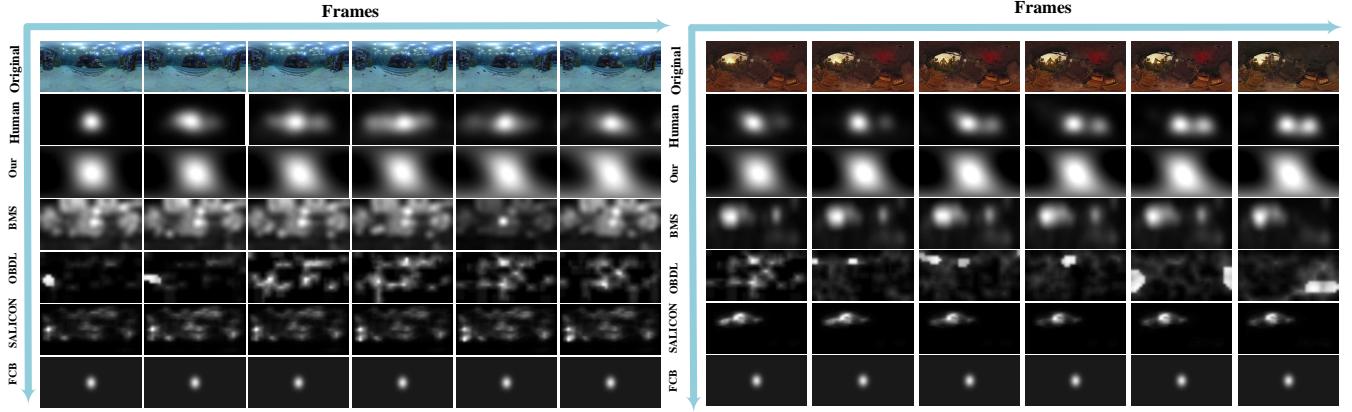


Fig. 10: HM maps of several frames selected from two test sequences in our PVS-HM database. They are all visualized in the 2D coordination. The second row shows the ground-truth HM maps, which are generated upon HM portions of all 56 subjects. The third to sixth rows show the HM maps of our, BMS [12] , OBDL [24],and SALICON approaches. The last row demonstrates the HM maps by the FCB baseline.

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

CC	Method	Average	KingKong	SpaceWar2	StarryPolar	Dancing	Guitar	PtSRun	InsideCar	RioOlympics	SpaceWar	CMLauncher2	Waterfall	Sunset	BlueWorld	Symphony	WaitingForLove
Non-FCB	Our	<b>0.808</b>	0.711	0.815	0.807	0.821	0.771	0.775	0.811	0.834	0.751	0.837	0.863	0.874	0.782	0.841	0.832
	BMS	0.328	0.007	0.091	0.496	0.233	0.528	0.276	0.481	0.289	0.305	0.364	0.479	0.548	0.177	0.287	0.363
	OBDL	0.183	-0.010	0.099	-0.018	0.323	0.124	0.200	0.337	0.285	0.053	0.170	0.052	0.470	0.018	0.291	0.348
FCB	Our	<b>0.609</b>	0.516	0.490	0.578	0.654	0.595	0.535	0.583	0.612	0.431	0.753	0.663	0.678	0.642	0.733	0.672
	BMS	0.578	0.463	0.450	0.495	0.601	0.587	0.430	0.568	0.607	0.392	0.746	0.653	0.646	0.651	0.713	0.675
	OBDL	0.448	0.280	0.362	0.382	0.478	0.264	0.297	0.517	0.487	0.330	0.714	0.661	0.463	0.567	0.397	0.521
	SALICON	0.466	0.142	0.260	0.454	0.377	0.478	0.398	0.538	0.461	0.347	0.640	0.523	0.595	0.614	0.572	0.592
FCB Only		0.616	0.517	0.495	0.580	0.655	0.591	0.533	0.589	0.621	0.441	0.768	0.676	0.694	0.649	0.745	0.686

front-center regions of panoramic video. Here, we model FCB by Gaussian distribution<sup>5</sup>, similar to the center bias of saliency detection. In the field of saliency detection, the center bias [2] is normally multiplied to saliency maps for improving saliency detection accuracy. Hence, we further report results of HM maps multiplied by FCB, for our and other approaches. 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 3 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. We can see from this table that our offline-DHP approach is drastically superior to other approaches, when FCB is not integrated. Specifically, our offline-DHP approach increases the CC value averaged over all 15 test sequences by XXX, XXXX and XXX, compared with OBDL, BMS and SALICON, respectively. Additionally, the increase of averaged NSS value is XXX, XXX and XXX in offline-DHP, over OBDL, BMS and SALICON. Once integrated with FCB, all four approaches have performance improvement, and our approach still performs best among all three approaches. Meanwhile, our approach significantly

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

outperforms the FCB baseline. 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.

Next, we move to the comparison of subjective results. Figure 10 shows several frames from two selected sequences and their ground-truth HM maps of human. In Figure 10, 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, as FCB can improve the performance of all four approaches in predicting HM maps (as seen in Table 3). From this figure, one can observe that HM maps of our approach are more close to the ground-truth HM maps, compared to other approaches and the FCB baseline. This indicates that our offline-DHP approach is capable of better locating HM positions of different subjects on panoramic video. Moreover, Figure 11 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 11 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.

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

NSS	Method	Average	KingKong	SpaceWar2	StarryPolar	Dancing	Guitar	BTSRun	InsideCar	RioOlympics	SpaceWar	CMLauncher2	Waterfall	Sunset	BlueWorld	Symphony	WaitingForLove
Non-FCB	Our	2.074	1.516	1.452	1.452	2.542	2.087	2.399	2.426	2.021	1.507	2.676	2.344	2.047	1.986	2.246	2.406
	BMS	0.950	-0.019	0.137	1.313	0.670	1.497	0.546	1.657	0.772	0.807	1.034	1.673	0.841	0.710	0.997	
	OBDL	0.457	-0.107	0.301	-0.006	0.980	0.393	0.215	1.375	0.637	0.064	0.660	0.073	1.015	0.035	0.260	0.964
	SALICON	0.620	-0.203	0.051	0.730	-0.236	0.965	0.337	1.823	0.600	0.344	0.921	0.410	0.669	1.138	0.456	1.298
FCB	Our	2.998	1.986	1.358	2.628	3.799	3.080	2.422	2.355	2.305	1.576	5.805	3.861	2.884	3.081	4.443	3.382
	BMS	2.802	1.529	1.109	2.621	3.068	3.079	1.573	2.267	2.308	1.472	5.465	4.014	2.815	3.127	4.229	3.356
	OBDL	1.965	0.580	0.916	1.618	1.947	0.834	0.564	2.138	1.947	1.100	5.080	4.060	1.814	2.572	1.465	2.844
	SALICON	2.169	0.019	0.448	2.411	0.641	2.117	1.728	2.013	1.776	1.380	5.333	3.009	2.674	2.909	2.780	3.292
FCB Only		2.130	0.927	0.634	2.300	2.023	1.744	1.063	1.172	1.416	1.090	5.736	3.122	2.259	2.125	3.994	2.348

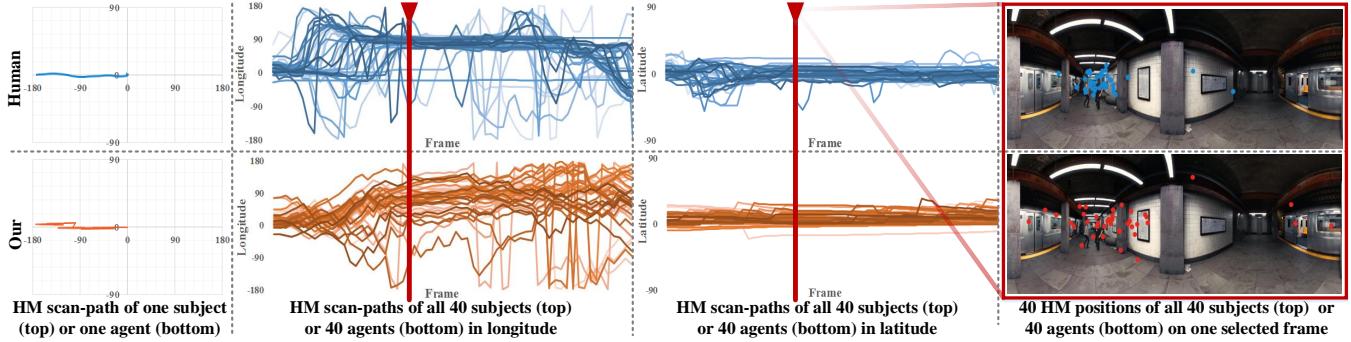
Fig. 11: Visualization in scan-paths and positions of HM generated by ground-truth and our offline-DHP approach, for sequence *Help*. The first column shows scan-path of one subject and one DRL workflow. The second and third columns show scan-paths of all 40 subjects and 40 DRL workflows, in longitude and latitude directions. The last column visualizes HM positions of ground-truth and our approach.

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 Polit	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	#DIV/0!
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

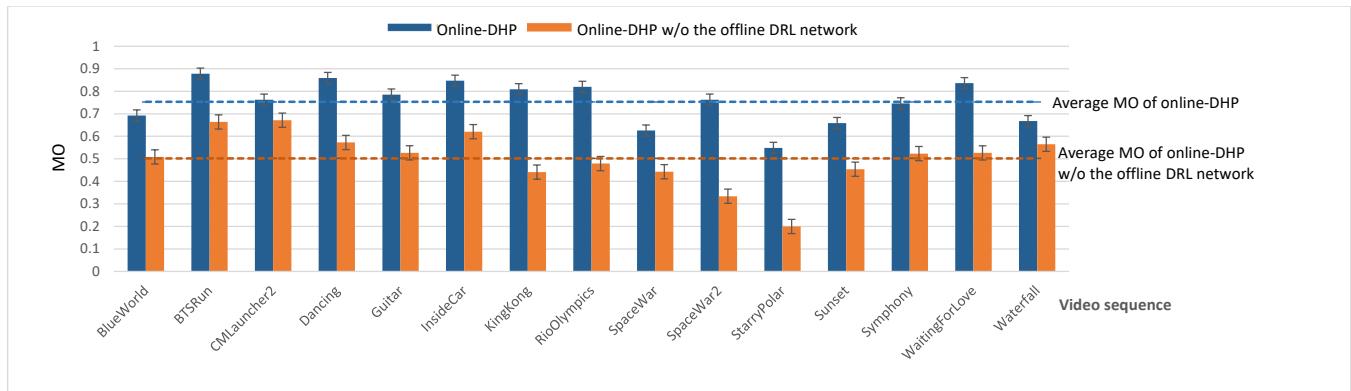


Fig. 12: MO results between Online-DHP and Online-DHP without offline-DRL network.

### 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 [17], 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 5*, 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 [17], MO of (6) is measured as the metric to evaluate the performance of online prediction for HM positions.

Table 5 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 5 that our online-DHP approach performs better than the deep 360 pilot [17], with a increase of XXXX in average MO. Such 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 [17] and two baseline approaches on the test set of [17]. 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 [17] and two baselines. In particular, our approach has XXX MO increase over [17], despite testing on the test set of [17]. 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 12 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 [17] and two baselines.

Table 5 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 [17]. 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 12 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.

## 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. Stengel and M. Magnor, "Gaze-contingent computational displays: Boosting perceptual fidelity," *IEEE SPM*, vol. 33, no. 5, pp. 139–148, 2016.
- [6] 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.
- [7] Y.-C. Su, D. Jayaraman, and K. Grauman, "Pano2vid: Automatic cinematography for watching 360-degree videos," in *ACCV*, 2016.
- [8] 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.
- [9] 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.
- [10] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE TIP*, vol. 13, no. 10, pp. 1304–1318, 2004.
- [11] 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.
- [12] J. Zhang and S. Sclaroff, "Exploiting surroundedness for saliency detection: a boolean map approach," *IEEE TPAMI*, vol. 38, no. 5, pp. 889–902, 2016.
- [13] 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.
- [14] Y. Liu, S. Zhang, M. Xu, and X. He, "Predicting salient face in multiple-face videos," in *CVPR*, 2017.
- [15] 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.
- [16] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [17] 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.
- [18] 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.
- [19] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vision research*, vol. 49, no. 10, pp. 1295–1306, Jun. 2009.
- [20] G. Bocciognone, "Nonparametric bayesian attentive video analysis," in *International Conference on Pattern Recognition (ICPR)*, 2008.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] S. Hossein Khatoonabadi, N. Vasconcelos, I. V. Bajic, and Y. Shan, "How many bits does it take for a stimulus to be salient?" in *CVPR*, 2015.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," *arXiv*, 2016.
- [29] Ç. Bak, A. Erdem, and E. Erdem, "Two-stream convolutional networks for dynamic saliency prediction," *arXiv*, 2016.
- [30] W. Wang, J. Shen, and L. Shao, "Deep learning for video saliency detection," *arXiv preprint arXiv:1702.00871*, 2017.
- [31] S. Minut and S. Mahadevan, "A reinforcement learning model of selective visual attention," in *AAMAS*, 2001.
- [32] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *NIPS*, 2014.
- [33] 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.
- [34] 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.
- [35] J. Foote and D. Kimber, "Flycam: Practical panoramic video and automatic camera control," in *ICME*. IEEE, 2000.
- [36] 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.
- [37] 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.
- [38] 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.
- [39] J. Li, C. Xia, Y. Song, S. Fang, and X. Chen, "A data-driven metric for comprehensive evaluation of saliency models," in *ICCV*, 2015.
- [40] E. Matin, "Saccadic suppression: a review and an analysis," *Psychological bulletin*, vol. 81, no. 12, p. 899, 1974.
- [41] M. J. Frederic P. Miller, Agnes F. Vandome, "Mean of circular quantities," *Mathematics*, 2010.
- [42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2014.
- [43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [44] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *AAAI*, 2015.
- [45] B. Shumaker and R. Sinnott, "Virtues of the haversine," *Sky and telescope*, vol. 68, pp. 158–159, 1984.
- [46] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3–4, pp. 279–292, 1992.
- [47] 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.



**Michael Shell** Biography text here.

PLACE  
PHOTO  
HERE

**John Doe** Biography text here.

**Jane Doe** Biography text here.