

# Predicting Head Movement in Panoramic Video: A Deep Reinforcement Learning Approach

Mai Xu<sup>✉</sup>, Senior Member, IEEE, Yuhang Song<sup>✉</sup>, Jianyi Wang<sup>✉</sup>, Minglang Qiao<sup>✉</sup>,  
Liangyu Huo<sup>✉</sup>, and Zulin Wang

**Abstract**—Panoramic video provides immersive and interactive experience by enabling humans to control the field of view (FoV) through head movement (HM). Thus, HM plays a key role in modeling human attention on panoramic video. This paper establishes a database collecting subjects' HM in panoramic video sequences. From this database, we find that the HM data are highly consistent across subjects. Furthermore, we find that deep reinforcement learning (DRL) can be applied to predict HM positions, via maximizing the reward of imitating human HM scanpaths through the agent's actions. Based on our findings, we propose a DRL-based HM prediction (DHP) approach with offline and online versions, called offline-DHP and online-DHP. In offline-DHP, multiple DRL workflows are run to determine potential HM positions at each panoramic frame. Then, a heat map of the potential HM positions, named 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 currently observed HM position, which is achieved by developing a DRL algorithm upon the learned offline-DHP model. Finally, the experiments validate that our approach is effective in both offline and online prediction of HM positions for panoramic video, and that the learned offline-DHP model can improve the performance of online-DHP.

**Index Terms**—Panoramic video, head movement, reinforcement learning, deep learning

## 1 INTRODUCTION

DURING the past years, panoramic video [1] has become increasingly popular due to its immersive and interactive experience. To achieve this immersive and interactive experience, humans can control the field of view (FoV) in the range of  $360^\circ \times 180^\circ$  by wearing head-mounted displays, when watching panoramic video. In other words, humans are able to freely move their heads within a sphere to make their FoVs focus on the attractive content (see Fig. 1 for an example). The content outside the FoVs cannot be observed by humans, i.e., not given any attention by the viewer. 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 modeling attention on panoramic video. In fact, human attention on panoramic video is composed of two parts: HM and eye fixations. HM determines FoV as the region to be seen in panoramic video through the position of HM sampled at each frame, called the HM position in this paper. Meanwhile, eye fixations decide which region can be captured at high resolution (i.e., fovea) within the FoV. Accordingly, HM prediction is the first step towards modeling human attention. Given the predicted HM, the

eye fixations within the FoV can be further estimated using the conventional saliency detection methods [2] for 2D video. The same as traditional 2D video, the attention model can be extensively utilized in many areas of panoramic video, such as region-of-interest compression [3], visual quality assessment [4], [5], rendering [6], synopsis [7], and automatic cinematography [8].

Unfortunately, few approaches have been proposed in modeling human attention on panoramic video, especially predicting the HM. Benefiting from the most recent success of deep reinforcement learning (DRL) [9], this paper proposes a DRL-based HM prediction (DHP) approach for modeling attention on panoramic video. The proposed approach applies DRL rather than supervised learning. It is because DRL maximizes the accumulated *reward* of the *agent's actions*, such that the predicted HM scanpaths can simulate the long-term HM behaviors of humans. In fact, HM prediction can be classified into two categories: offline and online prediction. In this paper, the offline HM prediction is used for modeling attention of multiple subjects on panoramic video, whereas the online prediction is used to predict the next HM position of a single subject, based on 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 offline-DHP and online-DHP, respectively. The codes for our offline-DHP and online-DHP approaches are downloadable from <https://github.com/YuhangSong/DHP>.

To our best knowledge, there exists no offline work to predict the 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

- The authors are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China.  
E-mail: {MaiXu, IceClearWJY, MinglangQiao, huoliangyu, wzulin}@buaa.edu.cn, yuhangsong2017@gmail.com.

Manuscript received 16 Oct. 2017; revised 14 June 2018; accepted 16 July 2018. Date of publication 23 July 2018; date of current version 10 Oct. 2019.  
(Corresponding author: Mai Xu.)

Recommended for acceptance by D. Crandall.

For information on obtaining reprints of this article, please send e-mail to: [reprints@ieee.org](mailto:reprints@ieee.org), and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TPAMI.2018.2858783

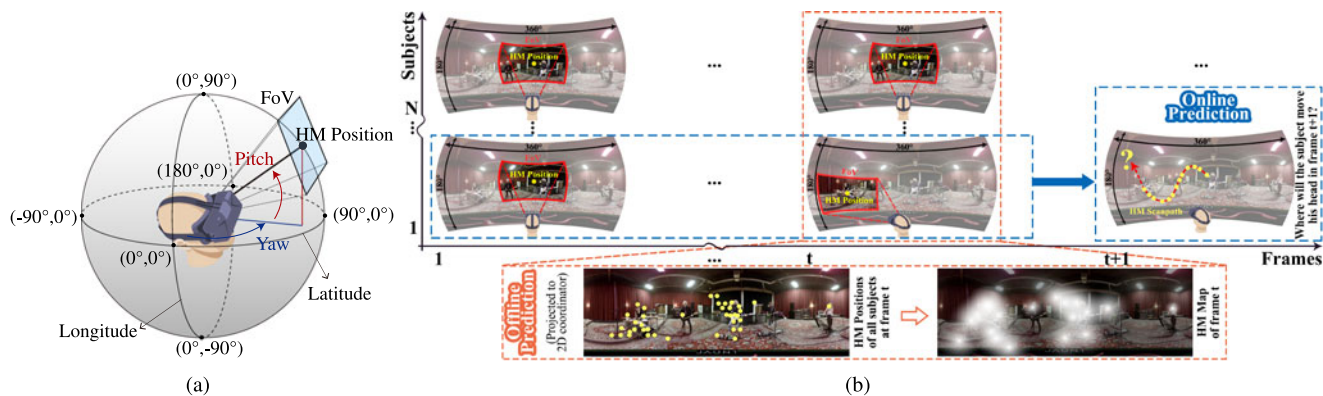


Fig. 1. (a) Illustration for head movement (HM) when viewing panoramic video. (b) Demonstration for FoVs and HM positions across different subjects. The heat map of HM positions from all subjects is also shown, which is defined as the HM map.

an image. Later, *Itti et al.* [11] proposed adding two features to [10], namely, 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 approach [12] and the Boolean map-based saliency model (BMS) [13]. Most recently, deep learning has been successfully applied in the works of saliency detection, such as SALICON [14] and Liu's approach [15]. Saliency detection differs from the offline prediction of HM positions in two aspects. (1) The input to saliency detection is 2D video in a plane, whereas panoramic video is a sphere (Fig. 1). Saliency detection can be applied to panoramic video that is projected from sphere to 2D plane, but projection normally causes distortion or content discontinuity, degrading the performance of predicting HM positions. (2) More importantly, saliency detection in 2D video assumes that humans are able to view all the content of each video frame. However, this assumption does not hold for panoramic video, as subjects can only see a limited range of the FoV at a single sight, rather than the full panoramic range of  $360^\circ \times 180^\circ$ .

In fact, different FoVs 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 in terms of HM positions. This finding is based on establishing and analyzing a new database, which consists of the HM data of 58 subjects 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 a sphere, and the positions in the HM maps are thus represented by the longitude and latitude in the geographic coordinate system (GCS) [17]. This paper visualizes the spherical HM maps by projecting them onto the 2D plane. The offline prediction of Fig. 1b 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 the HM positions with a 2D Gaussian filter.<sup>1</sup>

Specifically, our offline-DHP approach yields the HM maps of panoramic video via predicting the HM scanpaths of multiple *agents*, since subjects interactively control their HM along with some scanpaths according to video content. First,

we find from our database that the HM scanpaths of different subjects are highly consistent. Meanwhile, subjects are normally initialized to view the center of the front region in the beginning frames of panoramic video. Therefore, the HM positions at the subsequent frames can be yielded on the basis of the predicted scanpaths. Additionally, we find from our database that the magnitudes and directions of HM scanpaths are similar across subjects. In light of these findings, our offline-DHP approach models both the magnitudes and directions of HM scanpaths as the *actions* of multiple DRL *agents* and takes the viewed panoramic content as the *observation* of the *environment*. As such, the DRL model can be learned to predict HM positions. In training the DRL model, a *reward* is designed to measure the difference of *actions* between the DRL *agents* and subjects, indicating how well the *agents* imitate humans in terms of HM scanpaths. Then, the *reward* is optimized to learn the parameters in the DRL model. Given the learned model, the HM maps of panoramic video are generated upon the predicted HM positions, obtained from the scanpaths 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) when watching panoramic video. Specifically, the salient object is detected and tracked across panoramic video frames, via leveraging a region-based convolutional neural network (RCNN) [19] and recurrent neural network. Given the detected salient object and previous HM positions, the deep 360 pilot predicts to transit the 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 sports 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 in the online prediction of Fig. 1b). In this paper, we propose an online approach, namely online-DHP, to predict the HM positions on generic panoramic video. In contrast to [18], our online-DHP approach does not need to detect the salient object using the RCNN. Rather, our online-DHP approach 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* decides the *action* of the HM scanpath in the next frame, according to the ground-truth of the previous HM

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

scanpath and *observation* of video content. Consequently, the HM positions at the incoming frames can be predicted for our online-DHP approach.

This paper is the first attempt to apply the DRL algorithm in modeling human attention on panoramic video. The main contributions of this paper are three-fold:

- We establish a new panoramic video database that consists of HM positions of 58 subjects across 76 panoramic video sequences, with a thorough analysis of their HM data.
- We propose an offline-DHP approach to detect HM maps of panoramic video, and this approach 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 frame, based on the video content and HM scanpath till the current frame.

## 2 RELATED WORK

### 2.1 Saliency Detection

The only approach for predicting the HM positions of panoramic video is the most recent work of [8], in which Pano2Vid was proposed to obtain the FoV at each panoramic video frame. However, Pano2Vid primarily focuses on virtually generating a potential HM position at one frame, rather than modeling 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 the visual attention of humans on 2D video, by generating saliency maps of video frames. The studies on visual saliency began in 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. Subsequently, 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 video. Both [10] and [11] are heuristic approaches for detecting saliency, since they utilize the understanding of the human vision system (HVS) to develop the computational models. Recently, some advanced heuristic approaches, e.g., [12], [13], [20], [21], [22], [23], [24], [25], [26], have been proposed to detect saliency in 2D video. Specifically, [20] proposed a novel feature called *surprise*, which measures how the visual change attracts human observers, based on the Kullback-Leibler 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 the earth mover's distance to measure the center-surround difference in spatio-temporal receptive field, generating saliency maps for 2D video. Zhang et al. [13] explored the surround cue for saliency detection, by characterizing a set of binary images with random 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 these features 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], [32], [33], [34], [35], [36] have also been developed to detect 2D video saliency, rather than exploring the HVS-related features as in heuristic saliency detection approaches. These DNNs can be viewed as data-driven approaches. For static saliency detection, SALICON [14] fine-tuned the existing convolutional neural networks (CNNs), with a new saliency-related loss function. In [29], the architecture of multi-resolution CNN was developed for detecting saliency of images. In [35], a readout architecture was proposed to predict human attention on static images, in which both DNN features and low-level (isotropic contrast) features are considered. For dynamic saliency detection, [31] leveraged a deep convolutional 3D 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 using Gaussian mixture distribution. Similarly, Liu et al. [15] combined a CNN and multi-stream LSTM to detect saliency in video with multiple faces. Moreover, other DNN structures have been developed to detect either static saliency [27], [28], [30], [32] or dynamic saliency [31], [33], [34], [36].

Although saliency detection has been thoroughly studied in predicting eye movement in 2D video, no work has been developed to predict HM positions on panoramic video. Similar to saliency detection for 2D video, this paper proposes generating HM maps that represent the HM positions of multiple subjects. To obtain the HM maps of panoramic video, the HM positions are predicted by estimating the HM scanpaths of several *agents*. Similarly, in the saliency detection area, there exist some works [37], [38], [39], [40], [41], [42] that predict eye movement scanpaths for static images. In [37], a computational model was developed to simulate the scanpaths of eye movement in natural images. The proposed model embeds three factors to guide eye movement sequentially, including reference sensory responses, fovea periphery resolution discrepancy, and visual working memory. Sun et al. [38] proposed modeling both saccadic scanpaths and visual saliency of images, on the basis of super Gaussian component (SGC) analysis. Recently, data-driven approaches have been proposed to learn the scanpaths of eye movement in static images, such as the hidden Markov model in [39] and least-squares policy iteration (LSPI) in [40]. Most recently, deep learning has been utilized in [41], [42] for predicting the eye movement scanpaths in static images. However, to our best knowledge, there is no work on predicting the HM scanpaths on panoramic video.

In this paper, a DRL approach is developed for predicting the actions of the HM scanpaths from multiple *agents*. The actions are decided based on the environment of the panoramic video content, the features of which are automatically learned and then extracted by a DNN. Thus, our approach takes advantage of both deep learning and reinforcement learning, driven by the HM data of our panoramic video database. Note that although few works apply DRL to predict human attention, the attention model is widely used in the opposite direction to improve the performance of reinforcement learning, e.g., [43], [44], [45], [46].

### 2.2 Virtual Cinematography

Virtual cinematography of panoramic video, which directs an imaginary camera to virtually capture natural FoV, was



proposed in [8], [18], [47], [48], [49]. In general, virtual cinematography attempts to agree with the HM positions of humans at each panoramic video frame. The early work of [47] proposed cropping the object-of-interest in panoramic video, such that the natural FoV can be generated for virtual cinematography. Later, in [48], the cropped object-of-interest is tracked across frames by a Kalman filter, for automatically controlling the virtual camera in virtual cinematography of panoramic video. The approach of [48] can work on both compressed and uncompressed domains, because two methods were developed to detect the object-of-interest in compressed and uncompressed domains. The works of [47], [48] were both designed for the task of online virtual cinematography. These works can be considered 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 video. Specifically, Pano2Vid [8] learns to generate natural FoV at each panoramic frame. However, the learning mechanism of Pano2Vid is offline. In fact, natural FoV can be estimated at each frame in an online manner, which uses the observed HM positions of the previous frames to correct the estimation of natural FoV at the current frame. To this end, online virtual cinematography [18], [49] 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* that smoothly tracks the object-of-interest for panoramic video. In other words, the *agent* transmits the HM position across video frames to track the key object detected by the RCNN, given the observed HM positions at previous frames. Consequently, natural FoV can be generated online for automatically displaying the object-of-interest in virtual cinematography of panoramic video. In fact, object-of-interest tracking in panoramic video refers to continuously focusing and refocusing the intended targets. Both focusing and refocusing require a subject to catch up the object. Such a task is challenging in extreme-sports video, as the object-of-interest may be moving fast. Therefore, Lin et al. [49] 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], [47], [48], [49] all depend on the detector of the object-of-interest. Thus, they can only be applied in some specific panoramic video with salient objects, such as video conferencing or classroom scenes in [47], [48] and the sports video in [8], [18], [49]. 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 the object-of-interest. Consequently, our approach is object free, thus more suitable for generic panoramic video.

### 3 DATABASE ESTABLISHMENT AND FINDINGS

In this section, we collect a new database that includes 76 panoramic video sequences with the HM data of 58 subjects, called the PVS-HM database. Along with the HM data, the eye fixation data of 58 subjects are also obtained in our PVS-HM database. Our PVS-HM database allows

quantitative analysis of subjects' HM on panoramic video, and it can also be used for learning to predict where humans look at panoramic video. Our database is available at <https://github.com/YuhangSong/dhp> for facilitating 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 resolutions ranging from 3 to 8 K. As shown in Table 1 of the supplemental material, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2018.2858783>, the content of these sequences is diverse, including computer animation, driving, action sports, movies, video games, scenery, and so forth. Then, the duration of each sequence was cut to be from 10 to 80 seconds (averagely 26.9 seconds), such that fatigue can be reduced when viewing panoramic video. To ensure video quality, all panoramic video sequences were compressed using H.265 [50] without any change in 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, ranging in age from 18 to 36) wore the head-mounted display of an HTC Vive to view all 76 panoramic video sequences at a random display order. When watching panoramic video, the subjects were seated on a swivel chair and were allowed to turn around freely, such that all panoramic regions are accessible. To avoid eye fatigue and motion sickness, the subjects had a 5-minute rest after viewing each session of 19 sequences. With the support of the software development kit of the HTC Vive, we recorded the posture data of each subject as they viewed the panoramic video sequences. 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 GCS. In addition to the recorded HM data, the eye fixations were also captured by the VR eye-tracking module aGlass,<sup>2</sup> which was embedded in the head-mounted display of the HTC Vive.

Then, we mine our PVS-HM database to analyze the HM data of different subjects across panoramic video sequences. Specifically, we have the following five findings, the analysis of which is presented in the supplemental material, available online. 1) The HM positions on panoramic video possess front center bias (FCB). 2) When watching panoramic video, different subjects are highly consistent in HM positions. 3) The magnitude of HM scanpaths is similar across subjects, when viewing the same regions of panoramic video. 4) The direction of HM scanpaths on panoramic video is highly consistent across subjects. 5) Almost 50 percent subjects are consistent in one HM scanpath direction (among 8 uniformly quantized directions), and over 85 percent of subjects are consistent in three directions for HM scanpaths.

## 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. Fig. 2 shows the overall

2. When subjects viewing panoramic video, the aGlass device is able to capture the eye fixations within FoV at less than 0.5° error. See <http://www.aglass.com/?lang=en> for more details about this device.

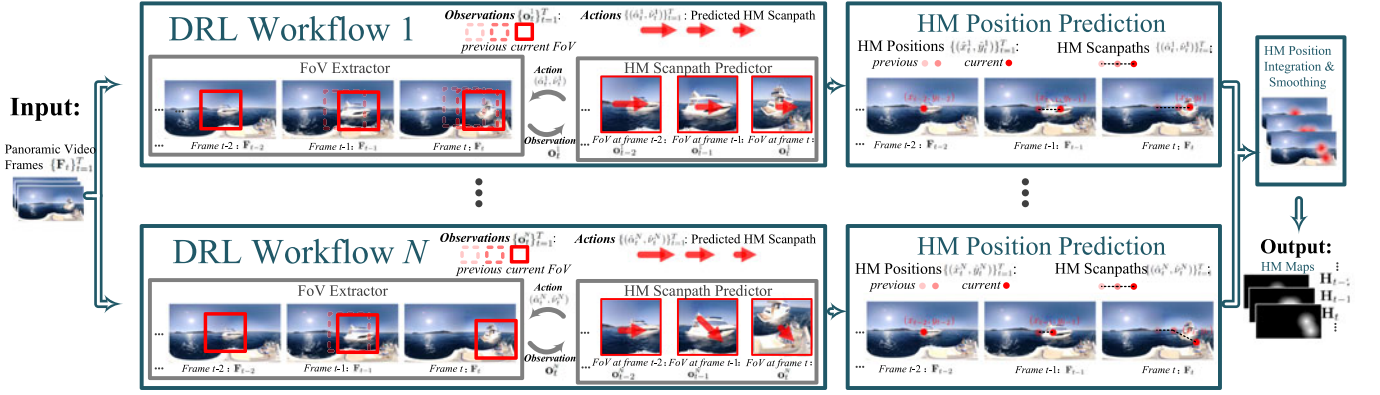


Fig. 2. Overall framework of the offline-DHP approach.

framework of our approach, in which the multiple DRL workflows are embedded to generate the HM maps of input panoramic video frames. The procedure and notations of Fig. 2 are presented in the following.

As shown in Fig. 2, the input to our offline-DHP approach is the panoramic video frames  $\{F_t\}_{t=1}^T$  with frame number  $t$  ranging from 1 to  $T$ . Since *Finding 2* has shown that the HM positions are highly consistent across different subjects, we propose to generate the HM maps  $\{H_t\}_{t=1}^T$  for modeling human attention on panoramic video, viewed as the output of our offline-DHP approach. The HM map  $H_t$  of frame  $t$  represents the probability of each pixel being the HM position. Assuming that  $\{(\hat{x}_t^n, \hat{y}_t^n)\}_{n=1}^N$  are the HM positions at the  $t$ th frame,  $H_t$  is obtained by convoluting  $\{(\hat{x}_t^n, \hat{y}_t^n)\}_{n=1}^N$  with a 2D Gaussian filter, similar to the saliency maps of 2D video. Here,  $n$  means the  $n$ th HM position and  $N$  is the total number of HM positions.

Because *Finding 5* has indicated that the HM scanpaths of different subjects are consistent in more than one direction, the HM positions  $\{(\hat{x}_t^m, \hat{y}_t^m)\}_{m=1}^M$  of subjects  $\{m\}_{m=1}^M$  may be different from each other. Accordingly, this paper assumes that the number of predicted HM positions  $N$  is equivalent to  $M$  at each frame, for predicting the HM positions of all subjects. In other words, to obtain  $(\hat{x}_t^n, \hat{y}_t^n)$ , our offline-DHP approach applies one DRL workflow to estimate the HM positions of one subject. Then,  $N$  DRL workflows are run to obtain  $N$  HM positions  $\{(\hat{x}_t^n, \hat{y}_t^n)\}_{n=1}^N$  at frame  $t$ , simulating the ground-truth HM positions of  $M$  ( $= N$ ) subjects at this frame. At a panoramic frame, each of the DRL workflows works independently to generate an HM position by randomly sampling actions based on a learned policy  $\pi_t$ , which is modeled as the predicted probability distribution of the HM direction at frame  $t$ . Note that all DRL workflows share the same policy  $\pi_t$  in our approach.

Let  $\hat{a}_t^n$  and  $\hat{v}_t^n$  be the direction and magnitude of the predicted HM scanpath at frame  $t$ , obtained from the  $n$ th DRL workflow. They are both viewed as the actions of the DRL workflow. In a single DRL workflow,  $\{(\hat{a}_t^n, \hat{v}_t^n)\}_{t=1}^T$  can be modeled by determining a series of actions:  $\{\hat{a}_t^n\}_{t=1}^T$  and  $\{\hat{v}_t^n\}_{t=1}^T$ . It is worth pointing out that  $\{\hat{a}_t^n\}_{t=1}^T$  and  $\{\hat{v}_t^n\}_{t=1}^T$  are predictable as the actions of the DRL workflow, since *Findings 3* and *4* have indicated that subjects are consistent in the magnitudes and directions of HM scanpaths. The direction and magnitude of the ground-truth HM scanpath are denoted by  $\alpha_t^m$  and  $v_t^m$  for the  $m$ th subject at frame  $t$ .

As can be seen in Fig. 2, in each workflow, one HM scanpath is generated through the interaction between the FoV extractor<sup>3</sup> and HM scanpath predictor. Specifically,  $\{\mathbf{o}_t^n\}_{t=1}^T$  denotes the FoVs of frames from 1 to  $T$  in the  $n$ th DRL workflow. Fig. 2 shows that FoV  $\mathbf{o}_t^n$  is extracted via making its center locate at the HM position  $(\hat{x}_t^n, \hat{y}_t^n)$ , in which  $(\hat{x}_t^n, \hat{y}_t^n)$  is generated by the predicted action of HM scanpath  $(\hat{a}_{t-1}^n, \hat{v}_{t-1}^n)$  at the previous video frame. Then, the content of the extracted FoV works as the observation of DRL, for predicting the next action of HM scanpath  $(\hat{a}_t^n, \hat{v}_t^n)$ . The HM scanpath generated by each DRL workflow is forwarded to obtain HM positions at incoming frames. Subsequently, the HM positions from multiple DRL workflows are integrated, and then smoothed by a 2D Gaussian filter. Finally, the HM maps  $\{H_t\}_{t=1}^T$  of the panoramic video are obtained, which model the heat maps for the HM positions at each frame.

## 4.2 DRL Model of the Offline-DHP Approach

As described in Section 4.1, the DRL workflow is a key component in our offline-DHP framework, which targets at predicting the HM scanpaths. This section presents how to train the DRL model of each workflow for predicting the HM maps. Note that our offline-DHP approach runs multiple workflows to train one global-shared model, the same as the asynchronous DRL method [9]. In this section, we take the  $n$ th workflow as an example. Fig. 3 shows the framework of training the DRL model. As shown in this figure, the FoV of the input video frame is extracted based on the action of the HM scanpath predicted at the previous frame. The extracted FoV, as the observation, is then fed into the DRL network. The structure of the DRL network follows [9], which has four 32-filter convolutional layers (size:  $21 \times 21$ ,  $11 \times 11$ ,  $6 \times 6$  and  $3 \times 3$ ), one flatten layer (size: 288) and LSTM cells (size: 256). Then, the 256-dimensional LSTM feature  $\mathbf{f}_t^n$  is output at frame  $t$ , as part of the observed state in the  $n$ th DRL workflow. In addition, the reward, which measures the similarity between the predicted and ground-truth HM scanpaths, 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 scanpath at the current frame. In this paper, we denote  $r_{n,t}^\alpha$  and  $r_{n,t}^v$  as the rewards for evaluating actions  $\hat{a}_t^n$  and

3. Note that the extracted FoV is  $103^\circ \times 60^\circ$ , which is the same as the setting of the head-mounted display.

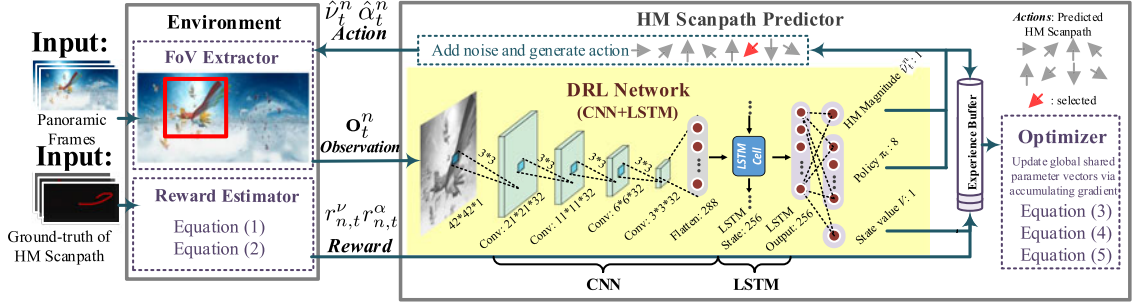


Fig. 3. Framework of training the DRL model to obtain each DRL workflow of the offline-DHP approach (Fig. 2).

$\hat{v}_t^n$ , respectively, in the  $n$ th DRL workflow. Finally, the *environment* of our DRL model is comprised by the *observation* of the extracted FoV and the *reward* of HM scanpath prediction.

In training the DRL model, the *environment* interacts with the HM scanpath predictor. The interaction is achieved in our DRL model through 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 onto the 2D region and is 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 scanpath predictor. In our work, the DRL network contains four convolutional layers and one LSTM layer [51], which are used to extract the spatial and temporal features, respectively. The details about the architecture of the DRL network can be found in Fig. 3.
- 3) At frame  $t$ , the DRL network produces the LSTM feature  $\mathbf{f}_t^n$ , HM scanpath magnitude  $\hat{v}_t^n$  and policy  $\pi_t$ . Here,  $\pi_t$  is modeled by the probability distribution over the *actions* of HM scanpath directions.
- 4) Given  $\pi_t$ , the HM scanpath predictor randomly samples an *action*  $\hat{a}_t^n$  with standard deviation  $\varepsilon$ , such that the exploration is ensured in decision making. Here,  $\hat{a}_t^n$  includes 8 discrete directions in GCS:  $\{0^\circ, 45^\circ, \dots, 315^\circ\}$ .
- 5) *Environment* is updated using  $\hat{v}_t^n$  and  $\hat{a}_t^n$ , 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 *observation*  $\mathbf{o}_{t+1}^n$  according to the HM position  $(\hat{x}_{t+1}^n, \hat{y}_{t+1}^n)$ . The *reward* estimator returns the *rewards*  $r_{n,t}^v$  and  $r_{n,t}^\alpha$  in predicting  $\hat{v}_t^n$  and  $\hat{a}_t^n$ , based on the ground-truth HM scanpaths of  $\{v_t^m\}_{m=1}^M$  and  $\{a_t^m\}_{m=1}^M$ .
- 6) A set of experiences  $\{\mathbf{o}_{t-1}^n, \mathbf{f}_{t-1}^n, \hat{v}_{t-1}^n, \hat{a}_{t-1}^n, r_{n,t-1}^v, r_{n,t-1}^\alpha\}$  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 termination condition of exceeding the maximum 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  $r_{n,t}^v$  in determining the *actions* of HM scanpaths. When training the DRL model, our goal is to make the prediction of  $\hat{a}_t^n$  and  $\hat{v}_t^n$  approach the ground-truth HM scanpaths. Thus, the *rewards*  $r_{n,t}^\alpha$  and  $r_{n,t}^v$  can be represented by the differences from  $\hat{a}_t^n$  to  $\{a_t^m\}_{m=1}^M$  and from  $\hat{v}_t^n$  to  $\{v_t^m\}_{m=1}^M$ , respectively. In our approach, these differences are measured by Gaussian

distributions. We further 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  $r_{n,t}^v$ , which are also modeled by the 2D Gaussian distribution. This consideration is because only the consistent HM regions have similar HM scanpaths, according to the analysis of *Finding 4*. Then,  $r_{n,t}^\alpha$  can be written as

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

In (1),  $D_d$  defines the phase difference, and  $D_s$  denotes the *great-circle distance* [52]. Moreover,  $\rho$  and  $\varrho$  are the standard deviations of Gaussian distributions, as the hyper-parameters.

In (1), the similarity score of  $e^{-\frac{1}{2} \left( \frac{D_d(\hat{a}_t^n, a_t^m)}{\rho} \right)^2}$  measures the similarity of HM direction between the ground-truth and *agent action*, while  $e^{-\frac{1}{2} \left( \frac{D_s((\hat{x}_t^n, \hat{y}_t^n), (x_t^m, y_t^m))}{\varrho} \right)^2}$  qualifies the validity of the corresponding similarity score in calculating the reward.

Then, given the HM direction, we can estimate its corresponding magnitude through reward  $r_{n,t}^v$ . Similar to (1), we have

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

where  $\varsigma$  is the hyper-parameter for the standard deviation of the HM scanpath magnitude. As defined in (2),  $e^{-\frac{1}{2} \left( \frac{\hat{v}_t^n - v_t^m}{\varsigma} \right)^2}$  is the similarity score of the HM scanpath magnitude. Reward  $r_{n,t}^v$  is valid in predicting the magnitude, only if both the predicted HM position and direction are similar to the ground-truth. Thus,  $e^{-\frac{1}{2} \left( \frac{D_d(\hat{a}_t^n, a_t^m)}{\rho} \right)^2}$  and  $e^{-\frac{1}{2} \left( \frac{D_s((\hat{x}_t^n, \hat{y}_t^n), (x_t^m, y_t^m))}{\varrho} \right)^2}$  are introduced in (2) to determine the validity of the similarity score.

**Optimization.** Next, we need to optimize the *rewards*  $r_{n,t}^\alpha$  and  $r_{n,t}^v$  when learning the network parameters of our DRL model in Fig. 3. Our offline-DHP approach applies 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  $\{\theta_v^n, \theta_\pi^n, \theta_V^n\}$ , producing  $\hat{v}_t^n$ ,  $\hat{a}_t^n$  and  $V$ . Here,  $V$  denotes the *state value* output by the DRL network, which is obtained using the same way as [9]. Meanwhile, global-shared parameter vectors  $\{\theta_v, \theta_\pi, \theta_V\}$ <sup>4</sup> are updated via an accumulating

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



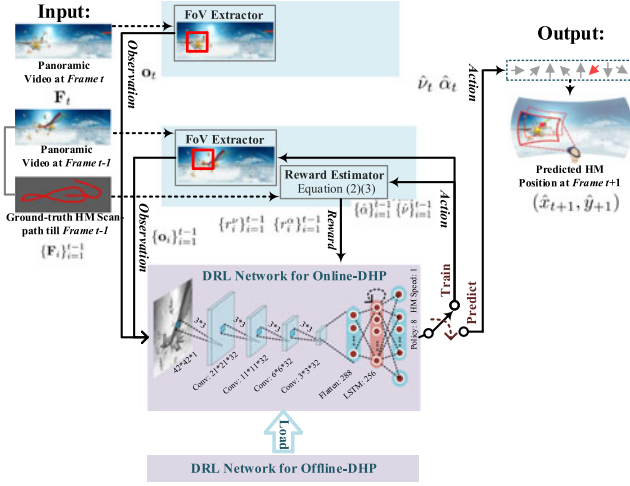


Fig. 4. Framework of the online-DHP approach.

gradient. For more details about the workflow-specific and global-shared parameter vectors, refer to [9]. In our approach, reward  $r_{n,t}^v$  is optimized to train  $\theta_v$  as follows

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

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

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

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

where  $\gamma$  is the discount factor of Q-learning [53]. In addition,  $V(\mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta_V^n)$  denotes state value  $V$  obtained by  $\mathbf{o}_t^n, \mathbf{f}_{t-1}^n$  and  $\theta_V^n$ ;  $\pi(\hat{\alpha}_t^n | \mathbf{o}_t^n, \mathbf{f}_{t-1}^n; \theta_\pi^n)$  stands for the probability of action  $\hat{\alpha}_t^n$  that is made by policy  $\pi$  from  $\mathbf{o}_t^n, \mathbf{f}_{t-1}^n$  and  $\theta_\pi^n$ . Finally, based on the above equations, RMSProp [54] is applied to optimize rewards in the training data. Consequently, the workflow-specific and global-shared parameter vectors can be learned to predict HM scanpaths. Finally, these learned parameter vectors can be used to determine the scanpaths and positions of HM through each DRL workflow in our offline-DHP approach.

## 5 ONLINE-DHP APPROACH

In this section, we present our online-DHP approach. The online-DHP approach refers to predicting 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 definitions of the notations in this section are similar to those in Section 4, and the only difference is that  $n$  and  $m$  are removed in all notations because there is only one subject/workflow in online-DHP. Additionally, we define the subject as the *viewer*, whose HM positions need to be predicted online. Fig. 4 shows the framework of our online-DHP approach. It is intuitive that the current HM position is correlated with the previous HM scanpaths and video content. Therefore, the input to our online-DHP framework is the *viewer's* HM scanpath  $\{(\alpha_1,$

$v_1), \dots, (\alpha_{t-1}, v_{t-1})\}$  and frame content  $\{F_1, \dots, F_t\}$ , and 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 viewed 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. In the first stage, the parameters of the DRL network are trained. In the second stage, the *action* of the HM scanpath is generated from the trained DRL network, to predict the HM position online. In the following, we discuss these two stages in more detail.

### 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 the front region, which is the general setting of the panoramic video player. Then, the trained DRL network of offline-DHP is loaded as the initial DRL network for online prediction, both sharing the same structure. The reason for loading the offline-DHP network is that it encodes the knowledge of HM-related features. Later, this initial DRL network is fine-tuned by the *viewer's* HM scanpath at incoming frames.

Next, we focus on the algorithm for training the DRL network in our online-DHP approach. As previously mentioned, 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 for 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,  $(\hat{\alpha}_i, \hat{v}_i)$  and  $(\alpha_i, v_i)$  are the predicted and ground-truth actions, respectively, of the HM scanpath for the *viewer*, and  $\mathbf{o}_i$  is the observation of the FoV content.
- 2) Take the action of  $(\hat{\alpha}_i, \hat{v}_i)$  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 one among 8 discrete HM scanpath directions, i.e.,  $\{0^\circ, 45^\circ, \dots, 315^\circ\}$ . The action of  $\hat{v}_i$  is a scalar of HM scanpath magnitude.
- 3) Calculate rewards  $(r_i^\alpha, r_i^v)$  from the reward estimator with (1) and (2), which measures how close the action  $(\hat{\alpha}_i, \hat{v}_i)$  is to the ground-truth HM scanpath  $(\alpha_i, v_i)$ . Here, the sums in (1) and (2) are not required for the reward calculation, since the ground-truth HM scanpath 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  $(\hat{\alpha}_i, \hat{v}_i)$ , and then input it to the DRL network.
- 5) Update the DRL network using (3), (4) and (5) and stop iterations, if the iteration number  $i$  is equivalent to  $t$ . Otherwise, proceed to step 2) for the next iteration.

Here, the definitions of *action*, *reward* and *observation* are the same as those in Section 4.2. The above iterations share the same implementation of training the DRL model in offline-DHP, which was 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 termination conditions. In our approach, there are two termination conditions. The first condition is the maximum number  $E$  of episodes. The second condition is based on the metric

of mean overlap (MO), which measures how close the predicted HM position is to the ground-truth HM position. MO ranges from 0 to 1, and a larger MO indicates a 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  $\text{FoV}_p$  and  $\text{FoV}_g$  represent the FoVs at the predicted and ground-truth HM positions, respectively. In (6),  $A$  represents the area of a panoramic region, which accounts for number of pixels. Then, the MO result of (6) at each episode is compared with a threshold  $th_{MO}$  to determine whether the training stage is terminated.

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

## 5.2 Stage II: Prediction

When the average MO is larger than threshold  $th_{MO}$ , the switch of Fig. 4 is turned to “predict”, and the DRL network makes an action of the HM scanpath at frame  $t + 1$ . Note that if the number of training episodes exceeds  $E$ , then the “predict” is also switched on, such that the training episodes end in a limited time. When entering the prediction stage, the DRL model trained in the first stage is used to produce the HM position as follows.

**Algorithm 1.** Algorithm for the Training Stage of Online-DHP to Predict the HM Position at Frame  $t + 1$

- 1: **Input:** Panoramic video frames  $\{F_1, \dots, F_t\}$ , and the ground-truth HM positions of the viewer  $\{(x_1, y_1), \dots, (x_t, y_t)\}$ .
- 2: Initialize the DRL network of online-DHP with parameter vectors  $\{\theta_v, \theta_\pi, \theta_V\}$ , by loading the network of offline-DHP.
- 3: **for**  $e = 1$  **to**  $E$  **do**
- 4: Initialize the HM position to be the center of the front region:  $\hat{x}_1 = 0, \hat{y}_1 = 0$ .
- 5: Initialize the LSTM feature to be the zero vector:  $\mathbf{f}_0 = \mathbf{0}$ .
- 6: **for**  $i = 1$  **to**  $t - 1$  **do**
- 7: Extract observation  $\mathbf{o}_i$  (i.e., FoV) from  $F_i$  according to  $(\hat{x}_i, \hat{y}_i)$ .
- 8: Obtain policy  $\pi_i$  and LSTM feature  $\mathbf{f}_i$  using the DRL network with  $\{\mathbf{o}_i, \mathbf{f}_{i-1}, \theta_\pi\}$ .
- 9: Select action  $\hat{\alpha}_i$  according to the  $\epsilon$ -greedy policy of  $\pi_i$ .
- 10: Generate action  $\hat{v}_i$  using the DRL network given  $\mathbf{o}_i, \mathbf{f}_{i-1}$  and  $\theta_v$ .
- 11: Calculate  $(\hat{x}_{i+1}, \hat{y}_{i+1})$  with regard to  $\hat{\alpha}_i, \hat{v}_i$ , and  $(\hat{x}_i, \hat{y}_i)$ .
- 12: Estimate rewards  $r_i^v$  and  $r_i^\alpha$  through (1) and (2) for  $(\hat{\alpha}_i, \hat{v}_i)$ .
- 13: Calculate the MO between  $(\hat{x}_i, \hat{y}_i)$  and  $(x_i, y_i)$ , denoted as  $MO_i$ .
- 14: Store a set of experiences:  $\{\mathbf{o}_i, \mathbf{f}_{i-1}, \hat{v}_i, \hat{\alpha}_i, r_i^v, r_i^\alpha\}$ .
- 15:  $i \leftarrow i + 1$ .
- 16: **end for**
- 17: Update  $\{\theta_v, \theta_\pi, \theta_V\}$  according to (3), (4), (5), in which  $\{\theta_v^n, \theta_\pi^n, \theta_V^n\}$  are replaced by  $\{\theta_v, \theta_\pi, \theta_V\}$ .
- 18:  $e \leftarrow e + 1$ .
- 19: Calculate the average MO through  $MO = \frac{\sum_{i=1}^{t-1} MO_i}{t-1}$ .
- 20: **if**  $MO > th_{MO}$  **then**
- 21: **break**
- 22: **end if**
- 23: **end for**
- 24: **Return:** The trained parameter vectors:  $\{\theta_v, \theta_\pi, \theta_V\}$ .

First, the LSTM features  $\{\mathbf{f}_i\}_{i=1}^{t-1}$  are sequentially updated from frame 1 to  $t - 1$ , based on the observed FoVs  $\{\mathbf{o}_i\}_{i=1}^{t-1}$  and the DRL parameters  $\theta_\pi$  of the training stage. Note that the LSTM feature is initialized with the zero vector  $\mathbf{0}$  at frame 1. Then,  $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_\pi\}$  produce action  $\hat{\alpha}_t$  of the HM scanpath direction. In addition, the HM scanpath magnitude  $\hat{v}_t$  is generated using  $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_v\}$ , in which the parameters of  $\theta_v$  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 scanpath  $(\hat{\alpha}_t, \hat{v}_t)$  at frame  $t$ . Algorithm 2 presents the summary of the prediction stage in online-DHP. Finally, online-DHP is achieved by alternating between the training and prediction stages until the currently processed frame.

**Algorithm 2.** Algorithm for the Prediction Stage of Online-DHP at Frame  $t + 1$

- 1: **Input:** The trained parameter vectors:  $\{\theta_v, \theta_\pi, \theta_V\}$  from the training stage, panoramic video frames  $\{F_1, \dots, F_t\}$ , and the ground-truth HM positions of the viewer  $\{(x_1, y_1), \dots, (x_t, y_t)\}$ .
- 2: Initialize the LSTM feature with the zero vector:  $\mathbf{f}_0 = \mathbf{0}$ .
- 3: **for**  $i = 1$  **to**  $t - 1$  **do**
- 4: Extract observation  $\mathbf{o}_i$  (i.e., FoV) from  $F_i$  according to  $(x_i, y_i)$ .
- 5: Obtain LSTM feature  $\mathbf{f}_i$  using the DRL network with  $\{\mathbf{o}_i, \mathbf{f}_{i-1}, \theta_\pi\}$ .
- 6:  $i \leftarrow i + 1$ .
- 7: **end for**
- 8: Extract observation  $\mathbf{o}_t$  (i.e., FoV) from  $F_t$  according to  $(x_t, y_t)$ .
- 9: Obtain policy  $\pi_t$  using the DRL network with  $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_\pi\}$ .
- 10: Choose action  $\hat{\alpha}_t$  using the greedy policy based on  $\pi_t$ .
- 11: Generate HM magnitude  $\hat{v}_t$  using the DRL network with  $\{\mathbf{o}_t, \mathbf{f}_{t-1}, \theta_v\}$ .
- 12: Estimate HM position  $(\hat{x}_{t+1}, \hat{y}_{t+1})$  at frame  $t + 1$ , upon  $\hat{\alpha}_t, \hat{v}_t$  and  $(x_t, y_t)$ .
- 13: **Return:** The HM position at frame  $t + 1$ :  $(\hat{x}_{t+1}, \hat{y}_{t+1})$ .

## 6 EXPERIMENTAL RESULTS

This section presents the experimental results for validating the effectiveness of our offline-DHP and online-DHP approaches. In Section 6.1, we discuss the settings of both offline-DHP and online-DHP in our experiments. Section 6.2 presents the results of ablation experiments. Sections 6.3 and 6.4 compare the performance of our offline-DHP and online-DHP approaches with those of other approaches in predicting HM positions, in the offline and online scenarios, respectively.

### 6.1 Settings

For evaluating the performance of offline-DHP, we randomly divided all 76 panoramic sequences of our PVS-HM database into a training set (61 sequences) and a test set (15 sequences). In training of the DRL model, the hyperparameters  $\rho, \varrho$  and  $\varsigma$  of (1) and (2) were tuned over the training set, when estimating the reward of HM scanpath prediction. As a result,  $\rho, \varrho$  and  $\varsigma$  were set to be 42, 0.7 and 1.0. In addition, we followed [9] to set the other hyperparameters of DRL. For example, we set 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 58, which is the same as the number of subjects in our PVS-HM database. Similar to [55], the HM positions predicted by the 58 DRL workflows were convoluted with a 2D Gaussian filter at each panoramic frame, to generate the HM map. In our experiments, the HM maps in a panorama were projected to a 2D plane for facilitating visualization. For evaluation, we measure the prediction accuracy of HM maps in terms of correlation coefficient(CC), normalized scanpath saliency (NSS), and area under receiver operating characteristic curve (AUC), which are three effective evaluation metrics [56] in saliency detection. Here, the shuffled-AUC is applied, in order to remove the influence of FCB in the evaluation. Note that larger values of CC, NSS and shuffled-AUC correspond to a more accurate prediction of HM maps.

For evaluating the performance of online-DHP, we compared our approach with [18] and two baseline approaches. The same as [18], MO of (6) is measured as the metric to evaluate the accuracy of online prediction in HM positions. Note that a larger value of MO means a more accurate online prediction in HM positions. Since the DRL network of offline-DHP was learned over 61 training sequences and used as the initial model of online-DHP, our comparison was conducted on all 15 test sequences of our PVS-HM database. In our experiments, the comparison was further performed over all test sequences of the database presented in [18], in order to test the generalization ability of our online-DHP approach. In our online-DHP approach, the hyperparameters of  $\rho$ ,  $\varrho$ ,  $\zeta$  and  $\gamma$  were set to be the same as those of the DRL workflows of offline-DHP. The other hyperparameters were identical to those in the most recent DRL work of [9]. In addition, the maximum number of episodes and the MO threshold were set to be 30 and 0.7, respectively, as the termination conditions in the training stage of online-DHP. Note that the MO threshold ensures the accuracy of HM position prediction, while the maximum episode number constrains the computational time of online-DHP.

## 6.2 Ablation Experiments

*Ablation on the Workflow Number in Offline-DHP.* Our offline-DHP approach generates the HM maps of panoramic video through the predicted HM positions of multiple workflows. Thus, we conducted the ablation experiments to investigate the performance of offline-DHP at different numbers of workflows. Fig. 5 shows the results of CC, NSS and shuffled-AUC for our offline-DHP approach, when the number of workflows  $N$  varies from 1 to 118. Note that the results in this figure are averaged over all 15 test panoramic sequences. We can see from Fig. 5 that CC approximately converges at  $N \geq 48$ , and that NSS and shuffled-AUC approximately converge at  $N \geq 58$ . Thus, we set the number of workflows  $N$  to be 58 in our experiments.

*Reinforcement Learning versus Supervised Learning.* Here, we evaluate the effectiveness of reinforcement learning applied in our approach by comparing with the supervised learning baseline. In the supervised learning baseline, the

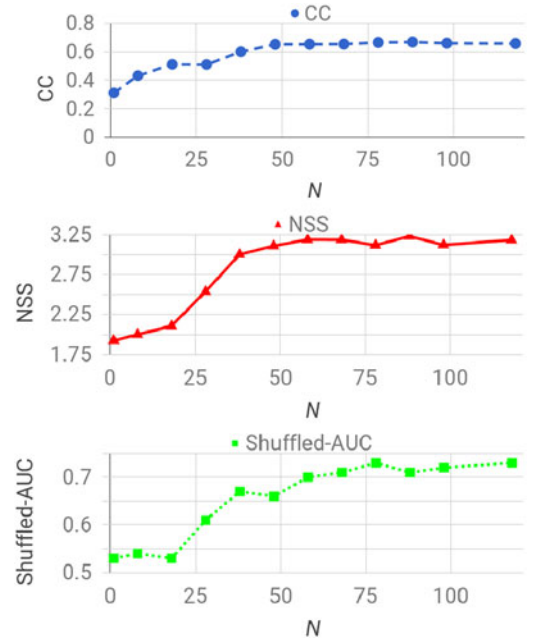


Fig. 5. Performance of offline-DHP at different numbers of workflows.

reinforcement learning component of our approach is replaced by a regressor and classifier. Specifically, the input to the supervised learning baseline is the FoV at the current frame, the same as our DHP approach. Then, the supervised learning baseline predicts the continuous magnitude of HM scanpath through a regressor. Additionally, the supervised learning baseline incorporates a classifier to predict the HM scanpath direction among 8 discrete directions in GCS:  $\{0^\circ, 45^\circ, \dots, 315^\circ\}$ . This ensures that the output of the baseline is the same as that of our DHP approach. For fair comparison, the DNN architecture of our DHP approach is used as the regressor and classifier, which have the same convolutional layers and LSTM cells as our approach. The magnitude regressor is trained by the MSE loss function, while the direction classifier is trained by the cross entropy loss function.

First, we compare the supervised learning baseline with our offline-DHP approach. To this end, the same as our offline-DHP approach, the supervised learning baseline runs 58 workflows to predict different HM positions for each panoramic frame. In each workflow, the baseline randomly samples one direction for the possible HM scanpath at each frame, according to the probabilities of directions by the trained classifier. Then, several HM positions are obtained upon the HM directions of all workflows, given the magnitude predicted by the trained regressor. Finally, the HM map is produced by convoluting these HM positions. Table 1 reports the CC, NSS and shuffled-AUC increase ( $\Delta$ CC,  $\Delta$ NSS and  $\Delta$ S-AUC) of our offline-DHP approach with the supervised learning approach as an anchor. We can see that the proposed offline-DHP approach performs much better against the supervised learning baseline. This validates the effectiveness of reinforcement learning applied in offline-DHP.

Second, we compare the supervised learning baseline with our online-DHP approach. The baseline predicts the HM position at the next frame using the trained magnitude regressor and direction classifier. In contrast, our online-DHP approach

TABLE 1  
 $\Delta$ ACC,  $\Delta$ NSS,  $\Delta$ S-AUC and  $\Delta$ MO between Offline-DHP/Online-DHP and the  
 Corresponding Supervised Baseline over 15 Test Sequences

|         |                | StarryPolar | Symphony | SpaceWar | RioOlympics | InsideCar | SpaceWar2 | Sunset | BlueWorld | Waterfall | Dancing | CMLauncher2 | Guitar | KingKong | BTSRun | WaitingForLove | Average |
|---------|----------------|-------------|----------|----------|-------------|-----------|-----------|--------|-----------|-----------|---------|-------------|--------|----------|--------|----------------|---------|
| Offline | $\Delta$ ACC   | -0.475      | -0.076   | 0.041    | 0.007       | 0.441     | 0.236     | 0.093  | 0.178     | 0.109     | 0.416   | 0.079       | 0.302  | 0.101    | 0.001  | -0.089         | 0.091   |
|         | $\Delta$ NSS   | -0.524      | 0.834    | 0.534    | 0.566       | 0.500     | 2.625     | 0.735  | 1.469     | 1.014     | 3.768   | 1.013       | 2.342  | 0.242    | 0.813  | 0.062          | 1.066   |
|         | $\Delta$ S-AUC | -0.025      | 0.024    | -0.025   | 0.156       | 0.330     | 0.112     | -0.056 | 0.113     | 0.025     | 0.267   | 0.255       | 0.040  | 0.133    | 0.320  | 0.101          | 0.118   |
| Online  | $\Delta$ MO    | 0.06        | 0.03     | 0.06     | 0.05        | 0.04      | 0.07      | 0.05   | 0.05      | 0.05      | 0.03    | 0.04        | 0.06   | 0.04     | 0.03   | 0.03           | 0.05    |

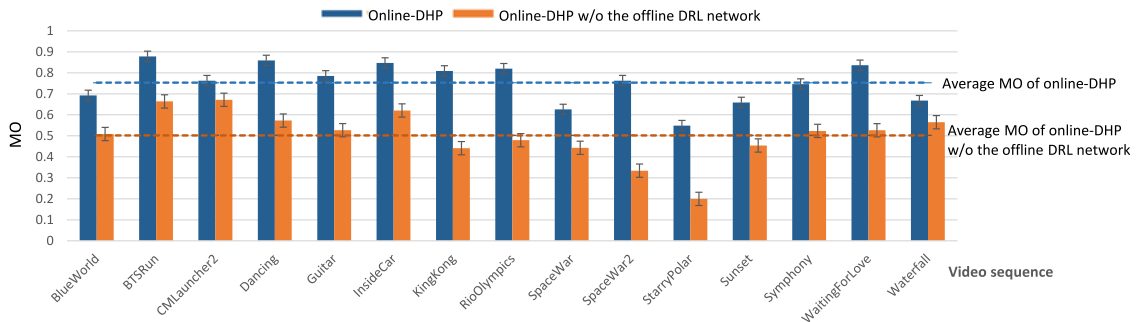


Fig. 6. MO results between the online-DHP approaches with and without the trained offline-DHP network.

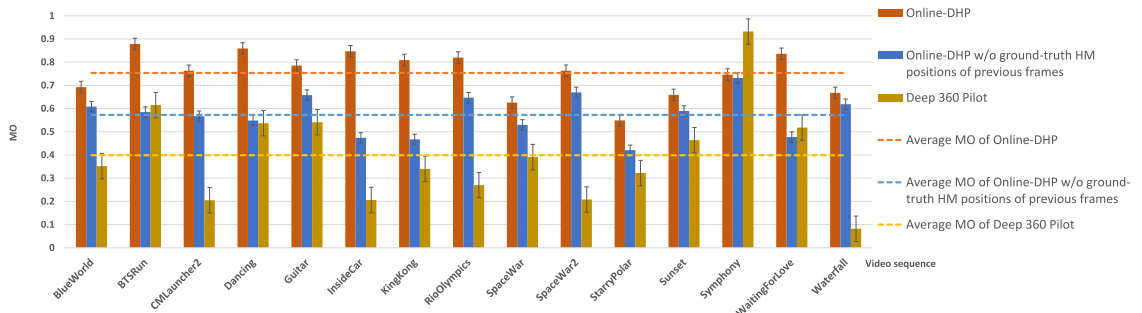


Fig. 7. MO results for Deep 360 Pilot, online-DHP approach, and online-DHP w/o ground-truth HM positions of previous frames.

predicts HM positions, based on reinforcement learning as introduced in Section 5. Table 1 tabulates the MO improvement ( $\Delta$ MO) of our online-DHP approach over the supervised learning baseline. As seen in this table, our online-DHP approach outperforms the supervised learning baseline in all sequences. Therefore, reinforcement learning is also effective in online-DHP.

*Influence of Offline DRL Network to Online-DHP.* It is interesting to analyze the benefits 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. Fig. 6 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 the MO results of our online-DHP approach, for all 15 sequences. In addition, the MO value can be increased from 0.50 to 0.75 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.

*Performance of Online-DHP w/o Previous Ground-Truth HM Positions.* For each test sequence, our online-DHP takes as input the ground-truth HM positions of previous frames to predict subsequent HM positions. The online-DHP approach belongs to online machine learning, and it is opposed to batch learning of deep 360 pilot [18], which generates the predictor by learning on the entire training dataset at once. Note that there is no online machine learning approach for predicting HM positions, and we can only compare with deep 360 pilot. For fair comparison with deep 360 pilot, Fig. 7 shows the results of our online-DHP approach using previous predicted HM positions as input, i.e., online-DHP w/o ground-truth HM positions of previous frames. As observed in Fig. 7, our online-DHP approach (MO = 0.57) performs considerably better than deep 360 pilot (MO = 0.40), when the previous ground-truth HM positions are not available in these two approaches for fair comparison. In addition, the ground-truth HM positions of previous frames can improve the performance of online-DHP, with MO increasing from 0.57 to 0.75 on average.

TABLE 2  
CC Results of Offline HM Map Prediction by Our and Other Approaches over 15 Test Sequences

| CC       | Method   | StarryPolar  | Symphony     | SpaceWar     | RioOlympics  | InsideCar    | SpaceWar2    | Sunset       | BlueWorld    | Waterfall    | Dancing      | CMLauncher2  | Guitar       | KingKong     | BTSRun       | WaitingForLove | Average      |
|----------|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------------|--------------|
| Non-FCB  | Our      | 0.185        | <b>0.710</b> | <b>0.573</b> | <b>0.717</b> | <b>0.783</b> | <b>0.673</b> | <b>0.673</b> | <b>0.678</b> | <b>0.763</b> | <b>0.837</b> | <b>0.585</b> | <b>0.645</b> | <b>0.751</b> | <b>0.764</b> | <b>0.471</b>   | <b>0.654</b> |
|          | BMS      | <b>0.450</b> | 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* | 0.168        | 0.216        | 0.106        | 0.189        | 0.292        | 0.291        | 0.235        | 0.255        | 0.393        | 0.281        | 0.220        | 0.365        | 0.217        | 0.285        | 0.288          | 0.253        |
| FCB      | Our      | 0.497        | <b>0.816</b> | <b>0.574</b> | <b>0.768</b> | <b>0.712</b> | <b>0.655</b> | <b>0.810</b> | <b>0.748</b> | <b>0.797</b> | <b>0.764</b> | <b>0.747</b> | <b>0.652</b> | <b>0.673</b> | <b>0.679</b> | <b>0.677</b>   | <b>0.704</b> |
|          | BMS      | <b>0.692</b> | 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  | 0.642        | 0.670        | 0.552        | 0.629        | 0.539        | 0.527        | 0.745        | 0.530        | 0.621        | 0.453        | 0.651        | 0.496        | 0.445        | 0.431        | 0.622          | 0.570        |
| 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        |

\*DNN based method has been fine-tuned by our database with their default settings.

### 6.3 Performance Evaluation on Offline-DHP

Now, we evaluate the performance of our offline-DHP approach in predicting the HM maps of all 15 test sequences from the PVS-HM database. To the best of our knowledge, there is no work on predicting the HM maps of panoramic video, and saliency prediction is the closest field. Therefore, we compare our offline-DHP approach to three state-of-the-art saliency detection approaches: OBDL [25], BMS [13] and SALICON [14], which are applied to panoramic frames mapped from sphere to plane using equirectangular projection. In particular, OBDL and BMS are the latest saliency detection approaches for videos and images, respectively. SALICON is a state-of-the-art DNN approach for saliency detection. For fair comparison, we retrained the DNN model of SALICON by fine-tuning over the training set of our database. Note that OBDL and BMS were not retrained because they are not trainable. In addition to the above three approaches, we also compare our approach to the FCB baseline, since *Finding 1* argues that human attention normally biases toward the front-center regions of panoramic video. Here, we model FCB using a 2D Gaussian distribution, similar to the center bias of saliency detection. Appendix 8 presents the details of the FCB modeling. In the field of saliency detection, the center bias [2] is normally combined with saliency maps to improve the saliency

detection accuracy. Hence, we further report the results of HM maps combined with the FCB feature, for our and other approaches. See Appendix A for more details about the combination of FCB.

Tables 2 and 3 tabulate the results of CC and NSS in predicting the HM maps of 15 test sequences, for our and other approaches. In these tables, the results of CC and NSS are averaged over all frames for each test sequence. As shown in these tables, when FCB is not integrated, our offline-DHP approach performs best among all three approaches and the FCB baseline, in terms of CC and NSS. More importantly, once integrated with FCB, all three approaches have performance improvement, and our approach still performs considerably better than other approaches. Specifically, our offline-DHP approach increases the average CC value by 0.242, 0.198 and 0.134, compared with OBDL, BMS and SALICON, respectively. Additionally, the increase of average NSS value is 1.245, 1.087 and 0.856 in our approach, in comparison with OBDL, BMS and SALICON. In a word, our offline-DHP approach is effective in predicting the HM maps of panoramic video, much better than other approaches and the FCB baseline.

Additionally, Table 4 compares the performance of our and other approaches in terms of shuffled-AUC. Note that

TABLE 3  
NSS Results of Offline HM Map Prediction by Our and Other Approaches over 15 Test Sequences

| NSS      | Method  | StarryPolar  | RioOlympics  | SpaceWar2    | Symphony     | SpaceWar     | Waterfall    | Sunset       | BlueWorld    | Guitar       | Dancing      | InsideCar    | CMLauncher2  | WaitingForLove | BTSRun       | KingKong     | Average      |
|----------|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------------|--------------|--------------|--------------|
| Non-FCB  | Our     | 0.899        | <b>2.806</b> | <b>2.237</b> | <b>3.346</b> | <b>2.180</b> | <b>3.765</b> | <b>2.529</b> | <b>3.196</b> | <b>3.461</b> | <b>5.297</b> | <b>4.402</b> | <b>3.529</b> | <b>2.278</b>   | <b>4.572</b> | <b>3.334</b> | <b>3.189</b> |
|          | BMS     | <b>1.313</b> | 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 | 0.628        | 0.584        | 0.396        | 1.093        | 1.348        | 1.528        | 1.194        | 0.877        | 1.167        | 1.541        | 0.876        | 1.265        | 0.858          | 1.121        | 1.362        | 1.056        |
| FCB      | Our     | 1.825        | <b>2.911</b> | <b>2.064</b> | <b>3.756</b> | <b>2.031</b> | <b>3.755</b> | <b>2.943</b> | <b>3.393</b> | <b>3.395</b> | <b>4.608</b> | <b>3.816</b> | <b>4.463</b> | <b>3.351</b>   | <b>3.931</b> | <b>2.883</b> | <b>3.275</b> |
|          | BMS     | <b>2.206</b> | 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 | 2.008        | 2.219        | 1.503        | 2.799        | 1.669        | 2.736        | 2.522        | 2.218        | 2.385        | 2.568        | 2.794        | 3.766        | 3.038          | 2.358        | 1.709        | 2.419        |
| 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        |



TABLE 4  
Shuffled-AUC Results of HM Map Prediction by Our and Other Approaches (without FCB) over 15 Test Sequences

| Method  | KingKong    | SpaceWar2   | StarryPolar | Dancing     | Guitar      | BTSRun      | InsideCar   | RioOlympics | SpaceWar    | CMLauncher2 | Waterfall   | Sunset      | BlueWorld   | Symphony    | WaitingForLove | Average     |
|---------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------|-------------|
| Our     | <b>0.72</b> | 0.63        | 0.46        | <b>0.82</b> | <b>0.73</b> | 0.84        | 0.80        | <b>0.69</b> | 0.60        | 0.76        | <b>0.72</b> | 0.64        | <b>0.70</b> | 0.70        | 0.68           | <b>0.70</b> |
| SALICON | 0.62        | 0.57        | 0.42        | 0.66        | 0.62        | 0.76        | 0.56        | 0.54        | <b>0.64</b> | <b>0.79</b> | 0.66        | 0.62        | 0.65        | 0.70        | 0.71           | 0.64        |
| BMS     | 0.65        | <b>0.65</b> | 0.43        | 0.74        | 0.54        | <b>0.88</b> | <b>0.83</b> | 0.53        | 0.62        | 0.63        | 0.66        | <b>0.70</b> | 0.49        | <b>0.77</b> | 0.69           | 0.65        |
| ODBL    | 0.70        | 0.60        | <b>0.55</b> | 0.81        | 0.68        | 0.86        | 0.62        | 0.68        | 0.57        | 0.56        | 0.47        | 0.69        | 0.47        | 0.75        | <b>0.74</b>    | 0.65        |

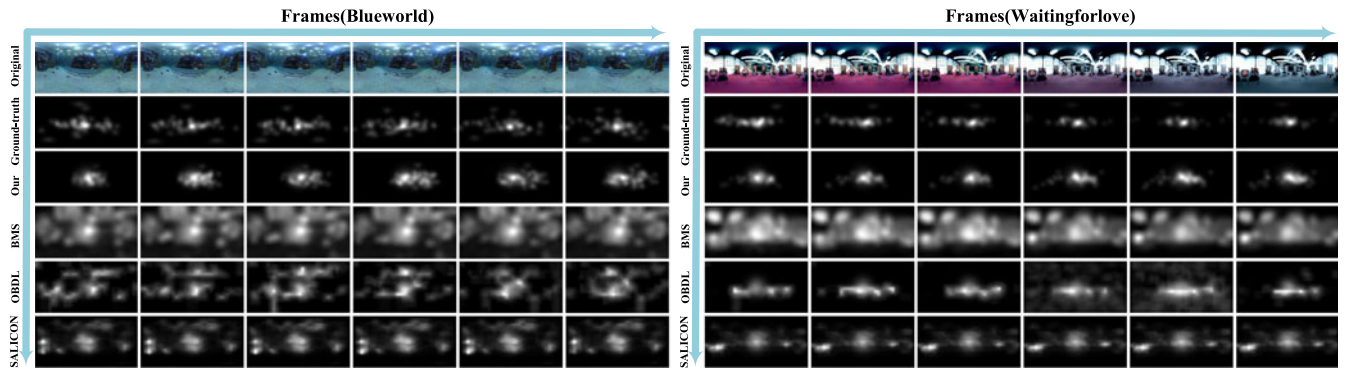


Fig. 8. HM maps of several frames selected from two test sequences in our PVS-HM database. They are all visualized in the 2D coordination. The second row shows the ground-truth HM maps, which are generated upon the HM positions of all 58 subjects. The third to sixth rows show the HM maps of our, BMS [13], ODBL [25], and SALICON [14] approaches.

TABLE 5  
MO Results of Online 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*        | <b>0.81</b> | <b>0.76</b> | <b>0.55</b> | <b>0.86</b> | <b>0.79</b> | <b>0.88</b> | <b>0.85</b> | <b>0.82</b> | <b>0.63</b> | <b>0.76</b> | <b>0.67</b> | <b>0.66</b> | <b>0.69</b> | 0.75        | <b>0.84</b>    | <b>0.75</b> |
| Deep 360 Pilot | 0.34        | 0.21        | 0.32        | 0.54        | 0.54        | 0.62        | 0.21        | 0.27        | 0.39        | 0.21        | 0.08        | 0.46        | 0.35        | <b>0.93</b> | 0.52           | 0.40        |
| Baseline 1*    | 0.20        | 0.21        | 0.16        | 0.22        | 0.20        | 0.21        | 0.22        | 0.20        | 0.21        | 0.21        | 0.20        | 0.20        | 0.21        | 0.20        | 0.21           | 0.20        |
| Baseline 2*    | 0.22        | 0.23        | 0.20        | 0.22        | 0.23        | 0.24        | 0.23        | 0.23        | 0.22        | 0.25        | 0.25        | 0.21        | 0.23        | 0.22        | 0.23           | 0.23        |

\*Both the online-DHP approach and baseline make prediction based on the ground-truth of previous frames.

FCB is not embedded in all approaches, since the shuffled-AUC metric is immune to FCB. In terms of the average shuffled-AUC, our approach has better performance than other approaches. This indicates that even not considering the influence of FCB, our approach again outperforms other approaches. It is worth mentioning that the shuffled-AUC of our offline-DHP approach ranks top in 6 out of 15 test sequences, while SALICON, BMS and ODBL have highest shuffled-AUC in 2, 5 and 2 sequences, respectively. The probable reasons are as follows. (1) In the evaluation, shuffled-AUC removes the influence of FCB, which can be learned by our offline-DHP approach. (2) The shuffled-AUC can be high even when the HM maps are non-sparse, i.e., far from ground truth. However, our approach yields more sparse HM maps than other approaches, close to ground-truth (see Fig. 8).

Next, we compare the subjective results. Fig. 8 shows several frames from two selected sequences and their ground-truth HM maps. In Fig. 8, we further visualize the HM maps generated by our and other approaches. Here,

the predicted HM maps are integrated with FCB, since the FCB feature can improve the performance of all three approaches (as presented in Tables 2 and 3). From this figure, one can observe that the HM maps of our approach are considerably closer to the ground-truth HM maps, compared with other approaches. This result indicates that our offline-DHP approach is capable of better locating the HM positions of different subjects on panoramic video.

#### 6.4 Performance Evaluation on Online-DHP

This section evaluates the performance of our online-DHP approach for predicting HM positions in the online scenario. The online scenario refers to predicting the HM position of one subject at each panoramic frame based on 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 in panoramic video. We also compare our online-DHP approach with two baselines. The first baseline

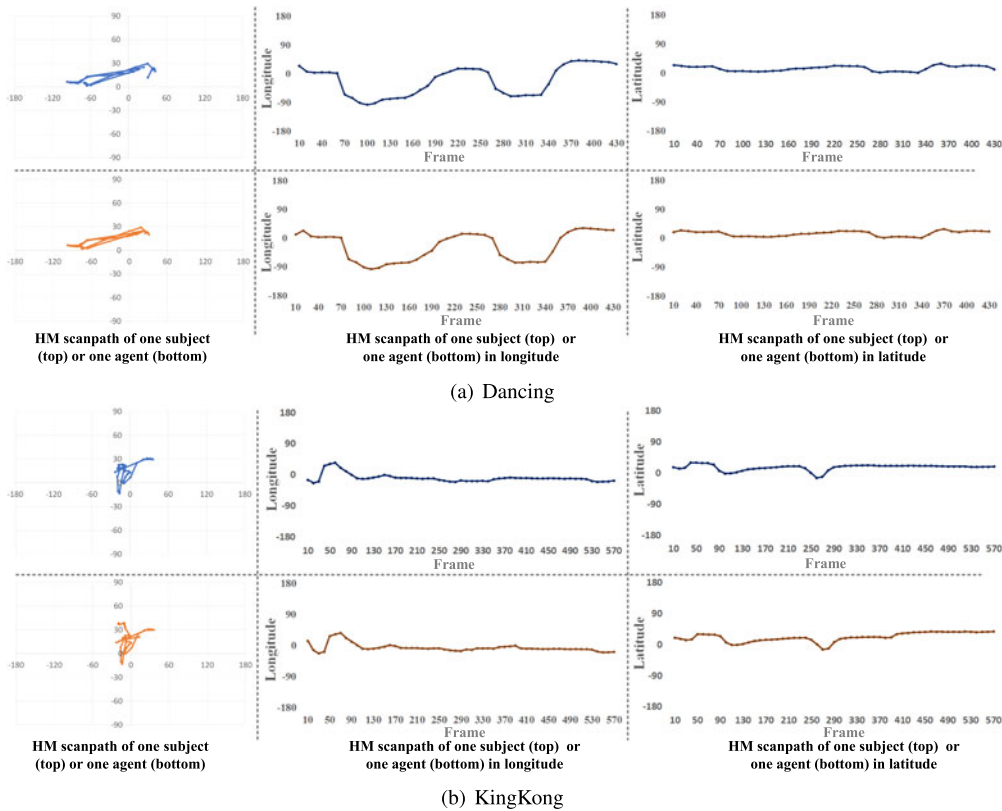


Fig. 9. Visualization in HM scanpaths generated by one subject and the online-DHP approach, for sequences *Dancing* and *KingKong*. Note that the HM scanpaths of one subject (among 58 subjects) are randomly selected and plotted, and then the corresponding HM scanpaths predicted by online-DHP are plotted.

(called baseline 1) keeps the HM scanpath of the current frame the same as that at the previous frame, such that the online HM position at each frame can be generated. The second baseline (called baseline 2) produces the HM positions, using the randomly generated HM scanpaths.

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 to predict HM positions online. Table 5 also shows that our online-DHP approach performs considerably better than the deep 360 pilot [18], with an increase of 0.35 in average MO. In addition, as shown in Table 5, our approach outperforms [18] over almost all sequences. The performance improvement of our approach is because (1) the online DRL model of our approach is capable of generating the accurate *actions* of HM scanpaths, and (2) the DRL network of offline-DHP is incorporated in our online prediction as the prior knowledge. Moreover, our approach is also effective for the generic panoramic sequences, while [18] fails in scenery panoramic video. For example, the MO result of [18] for the sequence *Waterfall* is 0.08, which is far less than 0.67 MO of online-DHP. This result is primarily because the deep 360 pilot [18] relies heavily on the object detection of RCNN.

Moreover, we visualize the ground-truth and predicted HM scanpaths, for subjective evaluation. Specifically, Fig. 9 plots the HM scanpaths by one subject and by the online-DHP approach, for the panoramic sequences of *Dancing* and *KingKong*. As shown in this figure, online-DHP is able

to obtain similar scanpaths as the subject, such that the HM positions can be accurately predicted online for each panoramic frame. In conclusion, our subjective evaluation, together with the above objective MO comparison, illustrates that the proposed online-DHP approach is effective in predicting the HM positions with the online manner.

To test the generalizability of our approach, we further evaluate the performance of our, the deep 360 pilot [18] and the two baseline approaches on the sports-360 dataset of [18]. For this 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 presented in Table 6. From this table, one may observe that our online-DHP approach again outperforms [18] and the two baselines, despite testing on the test set of [18]. In particular, the MO result of our approach is 0.90 for the panoramic sequences of dance, with 0.11 MO increase over [18]. Additionally, our approach improves the MO results of [18] by 0.07, 0.01, 0.01, and 0.10, for the sequences of skateboarding, parkour, BMX and basketball, respectively. In other words, the online-DHP approach is superior to the state-

TABLE 6  
MO Results for Online Prediction of HM Positions over the Sports-360 Dataset

| Method         | Skateboarding | Parkour     | BMX         | Dance       | Basketball  |
|----------------|---------------|-------------|-------------|-------------|-------------|
| DHP            | <b>0.78</b>   | <b>0.75</b> | <b>0.72</b> | <b>0.90</b> | <b>0.77</b> |
| Deep 360 Pilot | 0.71          | 0.74        | 0.71        | 0.79        | 0.67        |
| Baseline 1     | 0.15          | 0.17        | 0.16        | 0.17        | 0.17        |
| Baseline 2     | 0.22          | 0.19        | 0.18        | 0.22        | 0.18        |

of-the-art approach [18] in the online prediction of HM positions, over almost all classes of panoramic video sequences. Therefore, the generalization capability of our online-DHP approach can be confirmed.

## 7 CONCLUSION

In this paper, we have proposed the DHP approach for predicting HM positions on panoramic video. First, we established a new database named PVS-HM, which includes the HM data of 58 subjects viewing 76 panoramic sequences. We found from 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 an HM map, which encodes the possibility of each pixel being the HM position. Second, we proposed the offline-DHP approach to estimate HM maps in an offline manner. Specifically, our offline-DHP approach leverages DRL to make decisions on *actions* of HM scanpaths, via optimizing the *reward* of imitating the way that humans view panoramic video. Subsequently, the HM scanpaths of several *agents* from multiple DRL workflows are integrated to obtain the final HM maps. Third, we developed the online-DHP approach, which predicts the HM positions of one subject online. In online-DHP, the DRL algorithm was developed to determine the HM positions of one *agent* at the incoming frames, given the *observation* of previous HM scanpaths and the current video content. The DRL algorithm is based on the learned model of offline-DHP in extracting the spatio-temporal features of attention-related content. Finally, the experimental results showed that both 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 a 2D plane. Therefore, modeling attention on panoramic video is an important component in establishing human-like computer vision systems. It is an interesting future work to apply imitation learning for modeling attention on panoramic video. In particular, the *reward* of DHP may be learned from ground-truth HM data, belonging to inverse reinforcement learning that is a main category of imitation learning. Moreover, our work at the current stage mainly focuses on predicting HM positions, as the first step toward attention modeling of panoramic video. Future work should further predict eye fixations within the FoV regions of panoramic video. The potential applications of our approach are another promising work in future. For example, the online-DHP approach may be embedded in robotics, to mimic the way in which humans perceive the real world. Besides, panoramic video has large perceptual redundancy, since most of the panoramic regions cannot be seen by humans. It is thus possible to use the offline-DHP approach to remove such perceptual redundancy, and then the bit-rates of panoramic video coding can be dramatically saved.

## APPENDIX A

### ANALYSIS OF FCB COMBINED IN HM MAPS

The saliency detection literature [57] has argued that human attention has strong center bias in images or videos, and that the incorporation of center bias can improve the performance

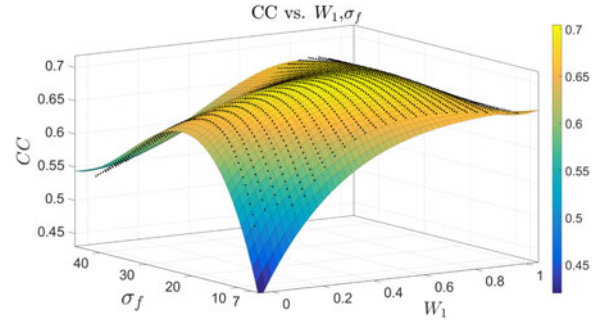


Fig. 10. The fitting surface of CC results between the predicted and ground-truth HM maps at various  $\sigma_f$  and  $w_1$ . The dark dots in this figure represent the CC results at each specific value of  $\sigma_f$  and  $w_1$ , which are used to fit the surface. Note that the CC results are obtained over all training data of the PVS-HM database.

of saliency detection. Similarly, FCB exists when viewing panoramic video, as discussed in *Finding 1*. Hence, this appendix presents the combination of the FCB feature and the offline-DHP approach. Here, we apply the FCB feature as an additional channel in generating the HM maps of panoramic video. Specifically, assume that  $\mathbf{H}^f$  is the HM map generated by the channel of the FCB feature. Similar to the center bias feature of image saliency detection [57], we apply the following 2D Gaussian distribution to model  $\mathbf{H}^f$  at 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  $(u, v)$  are the longitude and latitude of the GCS location in the map, and  $(u_f, v_f)$  are the longitude and latitude of the front center position in GCS. In addition,  $\sigma_f$  is the standard deviation of the 2D Gaussian distribution.

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

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

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

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

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 of CC over all training data of our PVS-HM database. Consequently, the optimal values of  $\sigma_f$ ,  $w_1$  and  $w_2$  are  $21.1^\circ$ , 0.48 and 0.52, respectively. These values are used to integrate the FCB feature in our offline-DHP approach. Note that the same way is applied to obtain the weights of  $w_1$  and  $w_2$ , when combining the FCB feature with other approaches.

Fig. 10 shows the results of CC between the predicted and ground-truth HM maps at various values of  $\sigma_f$  and  $w_1$ . From this figure, we can see that the CC results vary from 0.44 to



0.70 alongside the increase of  $w_1$  from 0 to 1, reaching the maximum value at  $w_1 = 0.48$  given  $\sigma_f = 21.1^\circ$ . 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 Fig. 10, at  $w_1 = 0.48$ , the CC value increases from 0.66 to 0.70, when  $\sigma_f$  grows from  $7^\circ$  to  $21.1^\circ$ , and then it decreases to 0.63 until  $\sigma_f = 43.6^\circ$ . Thus, the standard deviation of the 2D Gaussian distribution in (7) is set to be  $21.1^\circ$  for the FCB feature in our experiments.

## ACKNOWLEDGMENTS

This work was supported by the National Nature Science Foundation of China under Grant 61573037 and by the Fok Ying Tung Education Foundation under Grant 151061. M. Xu and Y. Song contributed equally to this work.

## REFERENCES

- [1] U. Neumann, T. Pintaric, and A. Rizzo, "Immersive panoramic video," in *Proc. 8th ACM Int. Conf. Multimedia*, 2000, pp. 493–494.
- [2] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, 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 Appl.*, vol. 76, pp. 5631–5659, 2017.
- [4] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1819–1831, Sep. 2016.
- [5] M. Xu, C. Li, Y. Liu, X. Deng, and J. Lu, "A subjective visual quality assessment method of panoramic videos," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2017, pp. 517–522.
- [6] M. Stengel and M. Magnor, "Gaze-contingent computational displays: Boosting perceptual fidelity," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 139–148, Sep. 2016.
- [7] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1971–1984, Nov. 2008.
- [8] Y.-C. Su, D. Jayaraman, and K. Grauman, "Pano2vid: Automatic cinematography for watching 360-degree videos," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 154–171.
- [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 *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
- [10] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.
- [11] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1304–1318, Oct. 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 Trans. Pattern Anal. Mach. Intell.*, 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 Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 889–902, May 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 262–270.
- [15] Y. Liu, S. Zhang, M. Xu, and X. He, "Predicting salient face in multiple-face videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3224–3232.
- [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 *Proc. Workshop Eye Tracking Vis.*, 2015, pp. 141–146.
- [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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1396–1405.
- [19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [20] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vis. Res.*, vol. 49, no. 10, pp. 1295–1306, Jun. 2009.
- [21] G. Boccignone, "Nonparametric Bayesian attentive video analysis," in *Proc. 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.
- [22] L. Zhang, M. H. Tong, and G. W. Cottrell, "SUNDAY: Saliency using natural statistics for dynamic analysis of scenes," in *Proc. Annu. Cogn. Sci. Conf.*, 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 Trans. Image Process.*, 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 Trans. Image Process.*, vol. 22, no. 8, pp. 3120–3132, Aug. 2013.
- [25] S. Hossein Khatoonabadi, N. Vasconcelos, I. V. Bajic, and Y. Shan, "How many bits does it take for a stimulus to be salient?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5501–5510.
- [26] M. Xu, L. Jiang, X. Sun, Z. Ye, and Z. Wang, "Learning to detect video saliency with HEVC features," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 369–385, Jan. 2017.
- [27] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2798–2805.
- [28] S. S. Kruthiventi, K. Ayush, and R. V. Babu, "DeepFix: A fully convolutional neural network for predicting human eye fixations," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4446–4456, 2017.
- [29] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 362–370.
- [30] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 825–841.
- [31] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," *5th Int. Conf. Learn. Representations*, 2017.
- [32] J. Pan, C. Canton, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. A. Giro-I Nieto, "SalGAN: Visual saliency prediction with generative adversarial networks," in *Proc. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1–9.
- [33] Ç. Bak, A. Erdem, and E. Erdem, "Two-stream convolutional networks for dynamic saliency prediction," *IEEE Trans. Multimedia*, vol. 20, pp. 1688–1698, 2018.
- [34] W. Wang, J. Shen, and L. Shao, "Deep learning for video saliency detection," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, 2017.
- [35] M. Kummerer, T. S. A. Wallis, L. A. Gatys, and M. Bethge, "Understanding low- and high-level contributions to fixation prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4799–4808.
- [36] S. Calderara, F. Solera, R. Cucchiara, A. Palazzi, and D. Abati, "Predicting the driver's focus of attention: The DR(eye)VE project," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1–25.
- [37] W. Wang, C. Chen, Y. Wang, T. Jiang, F. Fang, and Y. Yao, "Simulating human saccadic scanpaths on natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 441–448.
- [38] X. Sun, H. Yao, and R. Ji, "What are we looking for: Towards statistical modeling of saccadic eye movements and visual saliency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1552–1559.
- [39] H. Liu, D. Xu, Q. Huang, W. Li, M. Xu, and S. Lin, "Semantically-based human scanpath estimation with HMMs," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3232–3239.
- [40] M. Jiang, X. Boix, G. Roig, J. Xu, L. V. Gool, and Q. Zhao, "Learning to predict sequences of human visual fixations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1241–1252, Jun. 2016.
- [41] M. Assens, X. G. I. Nieto, K. McGuinness, and N. E. O'Connor, "SaltiNet: Scan-path prediction on 360 degree images using saliency volumes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2331–2338.
- [42] X. Shao, Y. Luo, D. Zhu, L. I. S. Li, and J. Lu, "Semantically-based human scanpath estimation with HMMs," in *Proc. Int. Conf. Neural Inf. Process.*, Dec. 2013, pp. 3232–3239.

- [43] S. Minut and S. Mahadevan, "A reinforcement learning model of selective visual attention," in *Proc. 5th Int. Conf. Auton. Agents*, 2001, pp. 457–464.
- [44] V. Mnih, N. Heess, A. Graves, et al., "Recurrent models of visual attention," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.
- [45] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.
- [46] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lanctot, and N. D. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.
- [47] J. Foote and D. Kimber, "FlyCam: Practical panoramic video and automatic camera control," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2000, pp. 1419–1422.
- [48] X. Sun, J. Foote, D. Kimber, and B. Manjunath, "Region of interest extraction and virtual camera control based on panoramic video capturing," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 981–990, Oct. 2005.
- [49] 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 *Proc. CHI Conf. Human Factors Comput. Syst.*, 2017, pp. 2535–2545.
- [50] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [51] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable mdps," in *Proc. Assoc. Advancement Artif. Intell.*, 2015, pp. 29–37.
- [52] B. Shumaker and R. Sinnott, "Virtues of the haversine," *Sky Telescope*, vol. 68, pp. 158–159, 1984.
- [53] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3/4, pp. 279–292, 1992.
- [54] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–30, 2012.
- [55] E. Matin, "Saccadic suppression: A review and an analysis," *Psychological Bulletin*, vol. 81, no. 12, 1974, Art. no. 899.
- [56] J. Li, C. Xia, Y. Song, S. Fang, and X. Chen, "A data-driven metric for comprehensive evaluation of saliency models," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 190–198.
- [57] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 2106–2113.



**Mai Xu** (M'10, SM'16) received the BS degree from Beihang University, in 2003, and MS degree from Tsinghua University, in 2006 and the PhD degree from Imperial College London, in 2010. From 2010–2012, he was working as a research fellow at Electrical Engineering Department, Tsinghua University. Since Jan. 2013, he has been with Beihang University as an associate professor. During 2014 to 2015, he was a visiting researcher of MSRA. His research interests mainly include image processing and computer vision. He has published more than 60 technical papers in international journals and conference proceedings, e.g., *IEEE Transactions on Image Processing*, CVPR and ICCV. He is the recipient of best paper awards of two IEEE conferences. He is a senior member of the IEEE.



**Yuhang Song** is working toward the PhD degree at the University of Oxford, in 2018. He is a research fellow from Beihang University. During 2017, he was a visiting researcher of Humanity-Centered Robotics Initiative, Brown University. His research interests mainly include reinforcement learning and related applications. He has published several technical papers in international conference proceedings, e.g., IEEE DCC.



**Jianyi Wang** received the BS degree in electrical engineering from Beihang University, in 2018 and is working toward the master's degree at Beihang University. He is a research assistant of Beihang University. He participated in lots of science and technology competitions and won many prizes during his undergraduate phase, e.g., 1st prize in "Challenge Cup" National Entrepreneurship Competition (The top entrepreneurship competition in China). Now his work is mainly related to panoramic video and reinforcement learning.



**Minglang Qiao** received the bachelor's degree in electrical engineering from Beihang University, Beijing, China, in 2018. He is currently working towards the master's degree at the MC2 Lab, Beihang University. His current research mainly focuses on saliency detection of images and panoramic videos. He has published papers in international conferences, e.g., ECCV.



**Liangyu Huo** received the bachelor's degree in electrical engineering from Beihang University, Beijing, China, in 2017. He is working toward the PhD degree at Beihang University. He participated in several science and technology competitions and won 1st prize in National College Student Information Security Contest. His research interests include multi-task reinforcement learning and hierarchical reinforcement learning.



**Zulin Wang** received the BS and MS degrees in electronic engineering from Beihang University, in 1986 and 1989, respectively, and the PhD degree at the same university, in 2000. He is currently a full professor at Beihang University, Beijing, China. He was the former dean of school of electronic and information engineering, Beihang University. His research interests include image processing and remote sensing technology. He is author or co-author of more than 100 papers and holds 6 patents, as well as published 2 books in these fields. He has undertaken approximately 30 projects related to image/video coding, image processing, etc.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).