Learning a Deep Agent to Predict Head Movement in 360-Degree Images

YUCHENG ZHU, GUANGTAO ZHAI, and XIONGKUO MIN, Shanghai Jiao Tong University
JIANTAO ZHOU, University of Macau

Virtual reality adequately stimulates senses to trick users into accepting the virtual environment. To create a sense of immersion, high-resolution images are required to satisfy human visual system, and low latency is essential for smooth operations, which put great demands on data processing and transmission. Actually, when exploring in the virtual environment, viewers only perceive the content in the current field of view. Therefore, if we can predict the head movements that are important behaviors of viewers, more processing resources can be allocated to the active field of view. In this article, we propose a model to predict the trajectory of head movement. Deep reinforcement learning is employed to mimic the decision making. In our framework, to characterize each state, features for viewport images are extracted by convolutional neural networks. In addition, the spherical coordinate maps and visited maps are generated for each viewport image, which facilitate the multiple dimensions of the state information by considering the impact of historical head movement and position information. To ensure the accurate simulation of visual behaviors during the watching of panoramas, we stipulate that the model imitates the behaviors of human demonstrators. To allow the model to generalize to more conditions, the intrinsic motivation is employed to guide the agent's action toward reducing uncertainty, which can enhance robustness during the exploration. The experimental results demonstrate the effectiveness of the proposed stepwise head movement predictor.

CCS Concepts: • **Computing methodologies → Model development and analysis**;

Additional Key Words and Phrases: VR, omnidirectional, 360 degree, panoramic, saliency, head movement prediction, deep reinforcement learning (DRL)

ACM Reference format:
Yucheng Zhu, Guangtao Zhai, Xiongkuo Min, and Jiantao Zhou. 2020. Learning a Deep Agent to Predict Head Movement in 360-Degree Images. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 4, Article 130 (December 2020), 23 pages.
https://doi.org/10.1145/3410455

This work was supported in part by the National Natural Science Foundation of China under grants 61901260, 61971476, 61831015, 61521062, and 61527804; in part by the Macau Science and Technology Development Fund under grants FDCT/022/2017/A1 and FDCT/077/2018/A2; and in part by the Research Committee at the University of Macau under grant MYRG2018-00029-FST.

Authors' addresses: Y. Zhu, G. Zhai (corresponding author), and X. Min, Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai, China; emails: {zyc420, zhaiguangtao, minxiongkuo}@sjtu.edu.cn; J. Zhou, State Key Laboratory of Internet of Things for Smart City, and Department of Computer and Information Science, University of Macau, Macau, China; email: jtzhou@umac.mo.

1551-6857/2020/12-ART130 $15.00
https://doi.org/10.1145/3410455

1 INTRODUCTION

Virtual reality (VR) is a powerful technology to enhance the expressiveness of traditional media, from images to movies to computer games. The immersive experience brought by VR gives people a sense of presence in the real world or artificial world. To achieve this goal, hardware components in VR system should produce stimuli to override human sense organs. The multidimensions of human perceptual experience increase the complexity of producing artificial simulations to all of the senses, so most VR systems only involve auditory and visual senses as a compromise. With the widespread availability of related electronics, more light-weight and low-cost headsets have appeared. The VR headset shows the trend of a high refresh rate, high resolution, and portability, and will maintain a progressive momentum, which will also encourage the development of more VR contents and applications. Actually, at any moment, only the part in the viewport will be rendered. Therefore, the utilization of eye-tracking technology and foveated rendering will greatly enhance the visual experience and will appear on the next-generation VR device [1]. The prediction of head and eye movements has become an important issue in VR development.

Among the efforts toward the saliency prediction for panoramas, head movement (HM) prediction is the most important yet daunting task that needs to be researched. HMs are important behaviors when viewers are using a VR headset. With HMs, users can orient the center of viewport to the position of interests and construct the percept by actively seeking new sensations. Accurate HM prediction can increase the bandwidth efficiency in the video streaming, where the agent can prioritize bandwidth for the video portions according to the probability of visual visits [2]. Furthermore, accurate prediction of HMs can increase the quality of experience since it would be possible to increase the visual quality and reduce the motion-to-photon delay by rendering the portions in the predicted viewports in advance before the head moves [3].

In the current research, most works only predict the saliency for panoramas (i.e., the distribution of viewports and fixations). Among these works, some algorithms are based on the human visual system (HVS) decomposition of visual signals, including the extraction of low-level and high-level features in the spatial domain and frequency domain to predict the saliency [4, 5, 6, 7, 8, 9, 10]. However, the complexity of HVS makes it difficult to depict which features are useful in the saliency prediction. Instead, convolutional neural networks (CNNs) can learn distinctive features between salient areas and non-salient areas [11, 12, 13]. However, the determination of the supervised learning (SL) model depends on the database. When the amount of annotated data is not enough for training, the performance of SL will degrade.

Some immersive databases have been established that collect head and eye movements data on immersive image/video [14–19]. However, their sizes are limited due to the strictness and complexness of data collections. To reduce the amount of required data, in the prediction of HM paths, deep reinforcement learning (DRL) can be employed to mimic the long-term HM behavior of the user [20]. Compared with SL, DRL can learn the optimal strategy by sampling actions and choosing the actions that maximize the reward. However, if the reward function is solely determined by the closeness of the prediction and the subject annotation, the same problem will arise. Therefore, in the design of reward function, the free-energy principle suggests that a psychovisual mechanism can guide the deployment of attention that can be employed as the guidance for the simulation of HM paths.

Some works have been proposed to explain in biological and physical sciences about human behaviors in perception (e.g., HM). Related works demonstrate the validity of the main tenet of the premotor theory of attention [21], which reveals that motor preparation and spatial attention employ the same neural substrates, and the intrinsic motivated actions are most likely to be those that canvass explicit data that are not liable to be predicted before the action [22]. In brief, the degree of discrepancy between the prediction by the generative model of the brain [23] and the actual external visual stimuli controls the salient action, which can also be defined as Bayesian surprise [24] and the expected free energy [25]. From the perspective of free energy, the principle suggests that the internal states of human agents are always at the low entropy level. This goal is realized by reducing the "surprise" under different environments that is upper bounded by a term called *free energy* [25]. Therefore, the minimization of surprise can be realized through reducing the free energy. More importantly, human agents can measure the free energy using the generative model based on the sensory information as the external state. In addition, human agents lower the free energy by actively seeking new sensations to eliminate the discrepancy between the external state and the inference of the internal generative model. Therefore, the perceptual inference builds the important intrinsic interactions between the visual perception and salient actions and guides the prediction of salient HMs in the VR system.

Operationally, our quest for improving the HM prediction for immersive images, which is guided by brain theory, is well realized by employing the psychovisual mechanism–based reward function in the framework of DRL. Specifically, the contributions of this work lie on several aspects. In the framework of DRL, our model extracts the features of the viewport images. In addition, the spherical coordinates of each pixel are provided as extra position information to enable the network to observe the position prior (e.g., center prior and equator bias). The visited map recording the HM positions before the current position is also included in the input to enable the network to learn the impact of past HMs. The psychovisual mechanism–based reward is incorporated in the reward to maximize the exploration and generalize the network. The psychovisual mechanism–based reward can measure the novelty of the state in terms of free energy and provide agents with bonus rewards whenever they visit a novel state. Therefore, some poorly understood states can be investigated and the agent's exploratory behavior can be enhanced. When making multi-step predictions, the previous predictions will be incorporated into the current input. The investigation of poorly understood states can reduce the propagation of discrepancy and potentially improve future performance. In addition, the psychovisual mechanism–based reward is much less sensitive to low-level visual features. Therefore, the policy equipped with a high-level psychovisual mechanism can also encourage the agent to have similar features even for some different visual signals, and help the model learn features that are more general. Given the learned model, the HM positions and path during the viewing of panoramas can be well predicted by the client in a stepwise way. Experiments that are conducted on the VR datasets named *Salient360* [14], *OIQA* [26] and *ODS* [15] demonstrate the effectiveness of our stepwise head movement predictor (SHMP).

The rest of this article is organized as follows. Section 2 presents some related works about saliency prediction models in panoramas. Section 3 presents implementation details of our model to predict head motions. In Section 4, the effectiveness of the model is proved by comparisons of the experimental results. Finally, concluding remarks are given in Section 5.

2 RELATED WORK

Recently, many databases have emerged for visual attention analysis and modeling on immersive images/videos. For establishing a visual attention database for immersive content, the motion sensors in HMD can record the HM data, and the eye tracker that is embedded into the HMD can capture the EM data. Some datasets provide both head and eye tracking data. Sitzmann et al. [15] found that starting points and watching conditions had an influence on the viewing behavior and designed the subjective experiments to record both HM and EM across 22 panoramas. Rai et al. [14] established a dataset of both head- and eye-tracking data from at least 40 subjects across 85 panoramas. In the work of Xu et al. [13], a new database was constructed. A total of 208 immersive videos and the corresponding HM and EM data were provided. Each video was viewed by at least 31 subjects. David et al. [19] constructed a database of immersive videos that recorded both HM and EM data across 19 videos. Some datasets only provide head-tracking data. Li et al. [17] constructed a database that contained 73 immersive videos and the HM data. Corbillon et al. [16] constructed a database that contained 7 omnidirectional videos and recorded the corresponding HM data. Ozcinar and Smolic [27] established a visual attention user dataset of 6 omnidirectional videos with different contents, recording viewport center trajectories of 17 participants. Fremerey et al. [18] constructed a dataset that recorded the head-tracking data of 48 subjects across 20 omnidirectional videos. In addition, the competition called *Salient360*[1] was held to understand how users watch and explore immersive content and model visual attention. We can see that several datasets have been established in recent years in this area. The collection of data in VR systems is more complex and time consuming, so the sizes of these datasets are limited.

Some saliency prediction models for immersive images and videos have been proposed. Handcrafted features can be used to predict the saliency. In these methods, the employed features are usually classified into top-down and bottom-up features. Rai et al. [5] designed several saliency weighting strategies and compared their performances to generate saliency maps from HM data. Zhu et al. [10] proposed a method to extend the existing saliency models designed for traditional images to panoramas. In addition, a model to predict HM, head-eye movement, and scanpath was also proposed. Abreu et al. [4] modified the current saliency models for traditional images to fast design omnidirectional image–oriented methods. Lebreton and Raake [9] modified existing saliency prediction models designed for traditional images to predict the saliency for omnidirectional images in the equirectangular format. Startsev and Dorr [6] employed various projection methods and compared the performances of the resulting saliency maps. Ling et al. [8] employed color dictionary–based sparse representation to predict the saliency for omnidirectional images. Battisti et al. [7] employed some low-level and semantic features to predict the saliency for omnidirectional images. Although the preceding models have tried to model visual attention, the features required to perform saliency prediction are all handcrafted and based on heuristics.

With the development of the CNN, many models use it to predict the saliency for 360-degree images. Because the omnidirectional images are distributed on the sphere, some methods mapped the images to the plane, then the CNN was employed to process the resulting 2D images [28, 29]. To enable the CNN to learn the spherical features, the spherical convolution was proposed by Su and Grauman [30] to extract features on panoramas. Cohen et al. [31] proposed spherical CNNs and employed spherical rotation-equivariant cross correlation to extract spherical features. To predict the saliency in panoramas, Monroy et al. [11] fed cube-mapped images and the location map into the designed network and trained the model in an end-to-end manner. Xu et al. [13] predicted the gaze in immersive videos by referring to the historical gaze path and employing the temporal and spatial saliency. Cheng et al. [12] employed cube padding and proposed a weakly supervised spatial-temporal network to predict saliency of immersive videos. However, these models are not comprehensive in terms of HM prediction, and it is difficult to conduct deep learning with limited data. Therefore, in this work, we devise the model that is based on DRL with designed reward to mitigate the problem.

Different from the preceding methods, here we employ the framework of DRL to predict the HM in immersive images. In the framework of DRL, our model extracts the features of the viewport images. In addition, the spherical coordinates of each pixel and the visited map recording the HM positions before current position are also incorporated. To maximize the exploration and generalize the network, we incorporate the psychovisual mechanism-based reward, which can measure the free energy of the state and advocate the state of large free energy value. With the psychovisual

[1] https://salient360.ls2n.fr/.

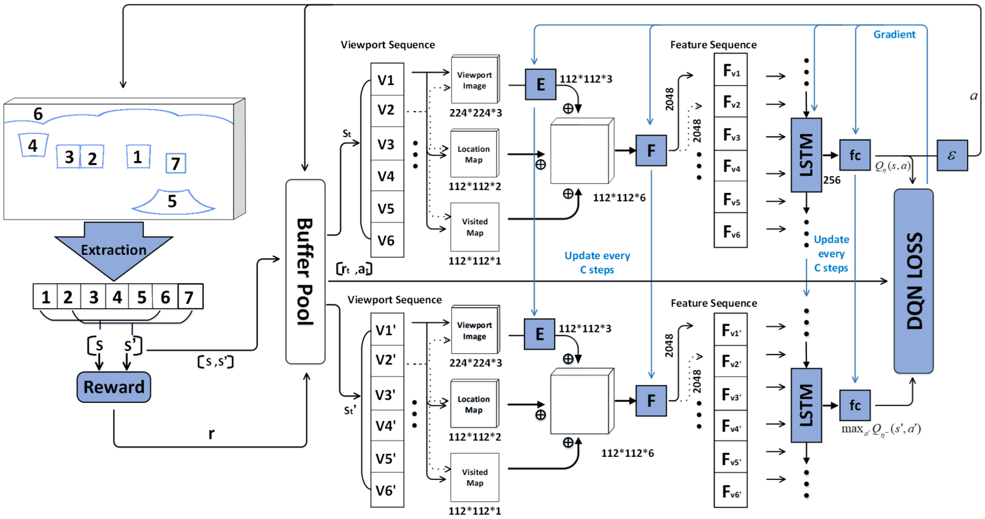


Fig. 1. In the framework, modules that need calculations are marked with blue. In the viewport extraction, the viewport image, location map, visited map, and saliency map are extracted at each viewport. A total of t_c ($t_c = 6$) [32] viewports constitute one state. With the action a, the agent makes the transition from s to s' and calculates the reward r of the transition. The buffer pool stores and randomizes the extracted viewports, the calculated rewards, and the actions. During learning, a minibatch of transitions (s_t, a_t, r_t, s_t') are extracted uniformly from the buffer pool. The Image embedding network (E) extracts features of the viewport image. Then the image features are concatenated with the location map and visited map. In addition, the concatenated features are embedded by the feature embedding network (F). The output of F is fed as the input of the LSTM network. The whole network ends with a 48-way fully connected layer with softmax. The output of the evaluation network and target network are used to calculate the loss, which is employed to update the evaluation network. The target network remains fixed between individual updates and is only updated every fixed step. The ε-greedy policy is employed to select the action.

mechanism-based reward, the agent is encouraged to investigate some poorly understood states and perform exploratory behavior. In addition, the psychovisual mechanism–based reward is high-level. The policy equipped with the psychovisual mechanism–based reward is less sensitive to low-level visual features, which can help the agent learn features that are more general and exploit the high-quality action under different conditions.

3 DESIGNING THE SHMP

3.1 The Overall Framework

The Markov decision process is employed to model the decision-making process. The switch of viewport with HM over time t is denoted as the stochastic process $\{u_t, t \in T\}$, of which the state space is the set of all possible HM locations. Considering the retention of memory of our brain, the process of HM is not memoryless. To satisfy the Markov property stating that the stochastic process should be memoryless, inspired by time-delay embeddings [33], we embed the dependent "memory" into the state.

In the process of watching panoramas, with the HMs, viewers are continuously being exposed to new content. When watching panoramas, the working memory will temporarily retain the visual information from the field of view, which is referred to as short-term retention. Compared with long-term memory, we believe that short-term memory is more important here.[2] The information kept in short-term memory will spontaneously decay over time. Therefore, the conditional probability distribution of the future state depends upon the present states and the past states in a short term. We can define a new process $\{s_t, t \in T\}$ by prolonging the state such that the new state of each random variable s_t consists of the current HM locations and the "history" over a time interval:

$$\mathrm{S} = H^{t_c}, \tag{1}$$

where S is the state space that is formed by the t_c-element Cartesian product on the set H, and H is the set that contains all possible field of view center positions. After the Cartesian product, each member in S contains t_c elements. By involving additional historical states, process $\{s_t, t \in T\}$ satisfies the Markov property.

The transition probability is influenced by a human agent who makes decisions to affect the evolution of the system over time. There is the finite set A that is composed of available actions. In the operation of HM prediction, the action set consists of movements of discrete distances and directions. There are 48 available actions that the agent can take. The detailed definition will be introduced in Section 4.1. On the basis of action set A, policy $\chi(a|s)$ specifies the probability of choosing action a when in state s. Therefore, the transition probability can be written as Equation (2):

$$p(s_{t+1}|s_t) = \sum_{a \in \mathrm{A}} \chi(a_t|s_t) p(s_{t+1}|s_t, a_t), \tag{2}$$

where the transition probability that the process moves into s_{t+1} is influenced by the chosen action a_t. Thus, we have the decision-making process:

$$\tau = s_0, a_0, s_1, a_1, \ldots, s_{t_0-1}, a_{t_0-1}, s_{t_0}. \tag{3}$$

In the decision-making process τ, the agent will start from the s_0 and take the action a_0 to switch to the s_1, and so on. Considering the Markov property, we can compute the probability of the decision process τ as Equation (4):

$$\begin{aligned} p(\tau) &= p(s_0, a_0, s_1, a_1, \ldots), \\ &= p(s_0) \prod_{t=0}^{t_0-1} \chi(a_t|s_t) p(s_{t+1}|s_t, a_t). \end{aligned} \tag{4}$$

The goal is to specify and optimize the policy $\chi(a_t|s_t)$ when the state transition function $p(s_{t+1}|s_t, a_t)$ is known. To formulate the objective function, the instant reward is defined as $r_t = r(s_{t-1}, a_{t-1}, s_t)$, and we have the discounted return of the process:

$$G(\tau) = \sum_{t=0}^{t_0-1} \gamma^t r_{t+1}, \quad \gamma \in [0, 1), \tag{5}$$

where γ is the discount factor. When γ is close to one, more long-term reward will be included in the discounted return. Otherwise, the short-term reward is emphasized when γ is close to zero. Then the weighted average of the discounted return of the trajectories can be calculated as Equation (6), and we can optimize the policy by maximizing the expected return:

$$J = E_{p(\tau)}(G(\tau)). \tag{6}$$

[2] The transfer of short-term memory to long-term memory should undergo the process of consolidation, involving rehearsal and meaningful association.

The expected return can be further expanded by rearranging Equation (6), and we have

$$E_{p(\tau)}(G(\tau)) = E_{p(s_0)}\left[E_{p(\tau\backslash s_0)}\sum_{t=0}^{t_0-1}\gamma^t r_{t+1}\right], \tag{7}$$

where the term in the square brackets is the value function $V^{\chi}(s)$ that measures the expected return starting form state s_0 and following policy χ. For convenience, we can use $\tau_{0:t_0}$ to represent trajectory $s_0, a_0, s_1, \ldots s_{t_0}$. In addition, $\tau_{0:t_0} = s_0, a_0, \tau_{1:t_0}$. The further expanding with respect to the next state can be conducted as follows:

$$\begin{aligned} V^{\chi}(s_0) &= E_{p(\tau_{0:t_0}\backslash s_0)}\sum_{t=0}^{t_0-1}\gamma^t r_{t+1} \\ &= E_{p(\tau_{0:t_0}\backslash s_0)}\left[r_1 + \gamma\sum_{t=1}^{t_0-1}\gamma^t r_{t+1}\right] \\ &= E_{a_0\sim\chi(a_0|s_0)}E_{s_1\sim p(s_1|s_0,a_0)}\left[r(s_0,a_0,s_1) + \gamma E_{p(\tau_{1:t_0}\backslash s_1)}\sum_{t=1}^{t_0-1}\gamma^t r_{t+1}\right] \\ &= E_{a_0\sim\chi(a_0|s_0)}E_{s_1\sim p(s_1|s_0,a_0)}\left[r(s_0,a_0,s_1) + \gamma V^{\chi}(s_1)\right] \\ &= E_{a_0\sim\chi(a_0|s_0)}\left[Q^{\chi}(s_0,a_0)\right], \end{aligned} \tag{8}$$

where the $Q^{\chi}(s_0, a_0)$ is the state-action value function. For the ease of representation, the current state and next state are denoted as s and s', and actions based on the current state and next state are denoted separately as a and a'. By substituting $V^{\chi}(s')$, we will have

$$\begin{aligned} Q^{\chi}(s,a) &= E_{s'\sim p(s'|s,a)}\left[r(s,a,s') + \gamma V^{\chi}(s')\right] \\ &= E_{s'\sim p(s'|s,a)}\left[r(s,a,s') + \gamma E_{a'\sim\chi(a'|s')}[Q^{\chi}(s',a')]\right]. \end{aligned} \tag{9}$$

The greedy policy can be employed to choose the action that maximizes the state-action value. In addition, when the transition probability is unknown, bootstrapping can be employed to form the model-free method and estimate the optimal action value function [34] by the following update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma\max_{a'}Q(s',a') - Q(s,a)\right), \tag{10}$$

where α is the learning rate in the Q-learning. Utilizing the representation power of the neural network, a network with parameter η can be used to approximate the action value. Similar to the Q-learning in Equation (10), the update of the network uses the following loss function:

$$L(s,a,s'|\eta) = \left(r + \gamma\max_{a'}Q_{\eta^-}(s',a') - Q_{\eta}(s,a)\right)^2, \tag{11}$$

where η are the parameters of the Q-evaluation network and η^- are the network parameters used to compute the target $r + \gamma\max_{a'}Q_{\eta}(s',a')$. To reduce the correlation between the evaluation value and target value, the parameters η^- remain fixed between individual updates and are only updated every fixed step [35]. In addition, there is strong correlation between consecutive samples that are not suitable for learning. Randoming samples can remove the correlations, so we realize the experience replay by storing the experience of agent in the buffer pool. During the learning process, minibatches of experiences are uniformly sampled at random from the buffer pool. To ensure adequate exploration of the state space, the action distribution is selected by the ε-greedy policy that selects a random action with probability ε and follows the greedy policy with probability 1-ε. Figure 1 shows the diagram of our framework.

Table 1. Image Embedding Network

Layer Name	Output Size	Kernel, Depth, Stride
Conv1	112×112	7×7, 64, stride 2
Conv2_x	56×56	3×3 max pool, stride 2 $\begin{bmatrix}1\times1, & 64\\ 3\times3, & 64\\ 1\times1, & 256\end{bmatrix} \times 3$
Conv3_x	28×28	$\begin{bmatrix}1\times1, & 128\\ 3\times3, & 128\\ 1\times1, & 512\end{bmatrix} \times 4$
Conv4_x	14×14	$\begin{bmatrix}1\times1, & 256\\ 3\times3, & 256\\ 1\times1, & 1024\end{bmatrix} \times 6$
Conv5_x	7×7	$\begin{bmatrix}1\times1, & 512\\ 3\times3, & 512\\ 1\times1, & 2048\end{bmatrix} \times 3$
DConv6	14×14	5×5, 512, stride 2
DConv7	28×28	5×5, 128, stride 2
DConv8	56×56	5×5, 32, stride 2
DConv9	112×112	5×5, 8, stride 2
Conv10	112×112	1×1, 3

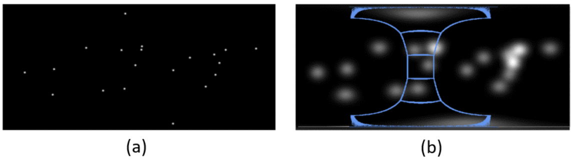


Fig. 2. Generation of a visited map. (a) The head location map. (b) Results after applying a Gaussian mask and the extraction of the visited map on different locations.

3.2 Network Architectures

To remove the correlation and improve the stability, we employ the experience replay and use a separate network for generating the targets as shown in Figure 1. The Q-network is mainly comprised of the image embedding module, feature embedding module, and sequence modeling module.

Image embedding. We use the image embedding network (E) to extract features from viewport images because Resnet can mitigate the degradation problem. In addition, Resnet can extract features with a large receptive field and meanwhile preserve fine details. We take the advantage of the Resnet50 [36] network that is pretrained on ImageNet [37]. We remove the average-pooling layer and fully connected layer in Resnet50. To upscale the feature map to the same dimensions as the location map, four learnable transpose convolution layers are employed to conduct the deconvolution. The final convolution layer is employed to map features to three channels. Table 1 shows more details.

Feature embedding. The location map, visited map, and viewport image features are combined as the input to the module of feature embedding (F). The two-channel location map records the longitude and latitude of each pixel in the viewport image, which provides the network the information of absolute position. The single-channel visited map records the visited information in the current viewport. As illustrated in Figure 2, we form the head location map by aggregating the

Table 2. Feature Embedding Network

Layer Name	Output Size	Kernel, Depth, Stride
Conv1	112×112	7×7, 64, stride 1
Conv2_x	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1{\times}1, & 64 \\ 3{\times}3, & 64 \\ 1{\times}1, & 256 \end{bmatrix} \times 3$
Conv3_x	28×28	$\begin{bmatrix} 1{\times}1, & 128 \\ 3{\times}3, & 128 \\ 1{\times}1, & 512 \end{bmatrix} \times 4$
Conv4_x	14×14	$\begin{bmatrix} 1{\times}1, & 256 \\ 3{\times}3, & 256 \\ 1{\times}1, & 1024 \end{bmatrix} \times 6$
Conv5_x	7×7	$\begin{bmatrix} 1{\times}1, & 512 \\ 3{\times}3, & 512 \\ 1{\times}1, & 2048 \end{bmatrix} \times 3$
Pool6	1×1	7×7, 2048, stride 1

historical visited locations. Considering the existing central fixation bias in scene viewing [38], we apply a Gaussian mask on the head location map on the sphere to get the global visited map [39]. The area of the current viewport on the global visited map is extracted as the visited map of the current viewport. The visited map provides the network the information of pseudo-visiting count and relative position. The concatenated features are fed to the next layers to refine the features with the extra information that is provided by the location map and visited map. The feature embedding network takes advantage of the plain network and shortcut connections in Resnet50 [36]. Table 2 shows more details.

Sequence modeling. Stacked Long Short-Term Memory (LSTM) is employed to model the sequence, which is comprised of two LSTM layers. Each LSTM cell at time t and level l has inputs $z(t)$ and hidden state $h(l, t)$. In the first layer, the input is the actual feature sequence from the feature embedding network and previous hidden state $h(1, t-1)$. In the second layer, the input is the hidden state of the corresponding cell in the previous layer $h(1, t)$ and previous hidden state $h(2, t-1)$. According to Miller [32], short memory has the capacity of about seven items. Thus, the model is designed to predict the current state based on six historical memory states (i.e., the number of timesteps is set as 6). At each timestep, the first layer obtains the feature sequence length of 2048 and a hidden size of 256, and the second layer obtains the sequence length of 256 and a hidden size of 256.

3.3 Designing Reward Function

We employ the intrinsic motivation to guide the agent's action toward reducing uncertainty. During the active visual perception [40] in the VR headset, human behavior and visual perception are tightly coupled. We move the head and eyes, and what we perceive in turn motivates our behaviors. Much evidence shows that the degree of discrepancy between the prediction by the generative model of the brain and the actual external visual stimuli, which is the cognitive dissonance in HVS and upper bounded by free energy [25], draws people's attention[22, 41]. Since the motor preparation and spatial attention employ the same neural substrates [21], the reduction of visually cognitive dissonance motivates the behavior.

The intrinsically motivated visual exploration is to maximally eliminate the discrepancy between the external state and the inference of internal generative model. Therefore, given a visual stimuli, or an image I, intrinsic motivation of visual exploration is positively correlated with the degree of discrepancy between the external visual stimuli and the prediction of the generative model. The unified internal generative model [25] can be employed to estimate the free energy, which is the upper bound of the discrepancy. The internal generative model for visual perception is parametric, which explains perceived scenes by adjusting the vector θ of parameters. To make this notion precise mathematically, we refer to the definition of free energy as in statistical physics and thermodynamics, which has also been widely used in visual quality measure [23, 42] and visual saliency prediction [41].

$$\begin{aligned} F(\theta) &= \int Q_{\mathcal{S}}(\theta|I) \log \frac{Q_{\mathcal{S}}(\theta|I)}{P_{\mathcal{G}}(\theta|I) P_{\mathcal{G}}(I)} d\theta \\ &= -\log P_{\mathcal{G}}(I) + \int Q_{\mathcal{S}}(\theta|I) \log \frac{Q_{\mathcal{S}}(\theta|I)}{P_{\mathcal{G}}(\theta|I)} d\theta \\ &= -\log P_{\mathcal{G}}(I) + KL\left(Q_{\mathcal{S}}(\theta|I) \| P_{\mathcal{G}}(\theta|I)\right) \end{aligned} \tag{12}$$

In brain theory [25], $P_{\mathcal{G}}(\theta|I)$ is referred to the recognition density that is conditioned on the internal generative model and external stimuli, and $Q_{\mathcal{S}}(\theta|I)$ is conditioned on the unconscious processing model that is instantly triggered by external visual stimuli (our brain digests what comes through the eyes using two sets of circuits) [43].

The $-\log P_{\mathcal{G}}(I)$ term is the log-evidence of the image data I given the model under the entire generative process, which will be reduced at each switch of viewport. The divergence term $KL\left(Q_{\mathcal{S}}(\theta|I) \| P_{\mathcal{G}}(\theta|I)\right)$ measures the discrepancy between the recognition density of the unconscious model and the internal generative model when perceiving a given scene. Therefore, the visual exploration is the process of updating and reducing the free energy, and the exploration of new content is better than the revisiting of the historical content. According to some instantiations [23, 42], the linear autoregressive (AR) model is the good choice for model $\mathcal{G}$. The AR model is defined as

$$x_i = \mathcal{X}^k(x_i)\boldsymbol{\alpha} + e_i, \tag{13}$$

where x_i denotes the pixel at location i, $\mathcal{X}^k(x_i)$ is the vector of k-member neighborhood of x_i, and $\boldsymbol{\alpha} = (a_1, a_2, \ldots, a_k)^T$ are the model parameters. The identity function is a simple choice for the unconscious processing model $\mathcal{S}$. Under a large sample limit, free energy equals the description length [44]. Therefore, the model can be estimated by minimizing the description length, and the free energy can be approximated as the total description length of the image. In general, we can form the training set $\mathcal{N}(x_i)$ in a neighborhood of x_i with $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ and the neighborhood member set with $\mathbf{X}(i,:) = \mathcal{X}^k(x_i)$. For a pixel x_i, we can deploy the k-th order piecewise AR model as in (13) on the training set and neighborhood member set. Under such conditions, to estimate $\boldsymbol{\alpha}$, the linear system can be written in matrix form as

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{X}\boldsymbol{\alpha}\|_2 . \tag{14}$$

The optimization on the linear system can be solved easily as $\boldsymbol{\alpha} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{x}$. With $\hat{\boldsymbol{\alpha}}$, we can then calculate the estimation error of a local pixel as

$$\hat{e}_i = x_i - \hat{x}_i = x_i - \mathcal{X}^k(x_i)\hat{\boldsymbol{\alpha}}. \tag{15}$$

Given the input image I, the point-wise error e_i can be computed and pooled to get the error map $E(I)$. The entropy of errors can be employed as the approximate to free energy term that is computed as $H(E) = \sum -P(e)\log P(e)$ with $P(e)$ the probability distribution of the errors.

To measure the elimination of the discrepancy, given the viewport images, the internal generative model $\mathcal{G}$ is optimized based on the visual perceptions that have already been observed, which is employed to measure the intrinsic motivation of next state. The algorithm for computing the

ALGORITHM 1: Estimation of Reward for Transition from s to s'

```
Input: Viewport image set $\mathbf{I} = \{I_1, I_2, \ldots I_n\}$ and the $\mathbf{I}' = \{I_2, I_3 \ldots I_{n+1}\}$,
       visited map $I^v_{n+1}$ and saliency map $I^s_{n+1}$
Output: The estimated reward $R$

for all images $I_l$ in $\mathbf{I} \cup \mathbf{I}'$ do
    for all pixel $x^l_i$ in $I_l$, pixel $x^t_i$ in $I_{n+1}$ do
        Form $\mathbf{x}$ and $\mathbf{X}$ for $x^l_i$
        model estimation $\hat{\boldsymbol{\alpha}}_i(x^l_i) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{x}$.
        error estimation $\hat{e}_{I_l \to I_{n+1}}(i) = x^t_i - \mathcal{X}^k(x^t_i)\hat{\boldsymbol{\alpha}}_i(x^l_i)$
    end for
    Pool $\hat{e}_{I_l \to I_{n+1}}$ as $E_{I_l \to I_{n+1}}$
    $F_{I_l \to I_{n+1}} = H(E_{I_l \to I_{n+1}})$,
end for
$R_f = \begin{cases} 0.5, & \text{if } \frac{1}{n+1}\sum_{I_l}\frac{F_{I_l \to I_{n+1}}}{F_{I_{n+1} \to I_{n+1}}} > \xi_1 \\ 0, & \text{otherwise.} \end{cases}$
form all-ones matrix $J_{w \times h}$
$Y_{n+1} = (J - I^v_{n+1}) \otimes I^s_{n+1}$
$R_s = \begin{cases} 0.5, & \text{if } \frac{1}{w \times h}\sum_{i,j} Y_{n+1}(i,j) > \xi_2 \\ 0, & \text{otherwise.} \end{cases}$
$R = R_f + R_s$
```

free-energy-based reward is summarized in Algorithm 1. Discrete rewards are used to drive the system away from bad states. And we can consider the relative sizes of the two components and scale the contributions of them expediently. The two parameters ξ_1, ξ_2 are set as 1.12 and 0.15, respectively. The parameters are selected from a coarse parameter sweep. For the unlabelled data, the calculation of free-energy-based reward need no observations of actions from human demonstrators. Therefore, reinforcement learning (RL) can be conducted in a self-supervised way.

4 PERFORMANCE EVALUATIONS AND ANALYSES

4.1 Implementation Details

Viewport determination. Research on human factors indicates that the optimal rotation angle of eyes downward and upward from normal line of sight (NLoS) is 15 degrees [45], and the maximum rotation angle of direct vertical viewing for eyes only is 20 degrees below NLoS and 40 degrees above NLoS. The optimum rotation angle of eyes to the left and right is also 15 degrees from NLoS, and the maximum rotation angle of direct horizontal viewing for eyes only is 35 degrees. For HM, the turning angle is ±65 degrees from NLoS to the left and right, 65 degrees above NLoS, and 35 degrees below NLoS. If both HM and EM are allowed, the horizontal field is ±95 degrees for head turning and eye rolling, and the vertical field can cover ±90 degrees. Figure 3 shows the details. Thus, in the implementation of our model, the size of the viewport is determined according to the rotation range of eyes. We use a local block image to simulate the viewport image and use the block-wise projection method that is proposed by Zhu et al. [10] to extract the local viewport.

HM discretization. The recordings of HM are the latitude and longitude over a continuous range (e.g., the range for latitude is $[0, \pi]$, and the range for longitude is $[0, 2\pi]$). The micro movements of head are messy and unpredictable. Many factors can result in micro changes of head position, such as the accompanying HM with the movement of body, the small movement of a headset because of the inertia and gravity, and the noise from the sensors. According to some research,

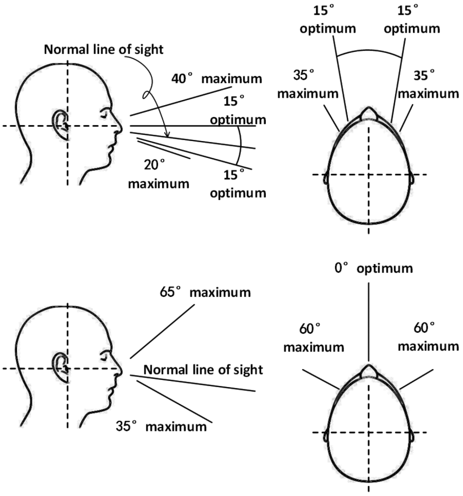


Fig. 3. Physiological functions of human vision (top row) and HM (bottom row).

small head position changes are confined because the eyes will counter-rotate substantially in the head direction [46]. In addition, in many cases, the goal-directed HMs will continue with the counter-rotation of eyes after the gaze is already on the target. Thus, it is reasonable to assume that the micro HMs are mostly from the aforementioned noises.

It is straightforward to regress the coordinates of the viewport in the geographic coordinate system [47]. But the even angular unit corresponds to the uneven length unit especially near the northern and southern poles, which means that near the poles, a small position change will cause a big longitude change. Therefore, the regression of coordinate is not suitable for panoramas. Actually, the transformation of the viewport can be characterized as the transformation of the fixed distance in the certain direction. But if the direction is regressed using the continuous radian, robustness is lacking when the transformation distance is small, because in such cases a small disturbance will cause a large change of the direction. To increase the robustness, the discretization of both transformation distance and direction will be a good choice. Another benefit of the discretization is that the number of training data with different labels can be easily controlled to make the training data balanced.

After the discretization on both the distance and direction, there are 48 different actions that the agent can take. Specifically, the current center of the viewport can be assigned as the north pole of the geographic coordinate system [47]. The polar coordinate is defined, in which viewport center V is the origin, latitude (ϖ) is the radial coordinate that measures the transformation distance, and longitude (ψ) is the angular coordinate that measures the transformation direction. As illustrated in Figure 4, in the polar coordinate, we can define the active area $a(\varpi, \psi|V)$ that satisfies $15^\circ < \varpi < 40^\circ$. Then the active area is segmented into 48 sectors, and each sector is associated with one cluster center. There are 16 innermost sectors that have the angle of $\frac{1}{8}\pi$ in radian and 32 outmost sectors that have the angle of $\frac{1}{16}\pi$ in radian. The latitude of sector arcs and cluster centers are set as $\varpi = 15^\circ, 18^\circ, 25^\circ, 32^\circ,$ and 40° separately. This allocation can ensure that most HMs locate in the two rings, and each sector has similar area. The locations of clusters are determined according to the rotation range of the human head. The cluster centers are the candidates for the next viewport center. Therefore, there are 48 available actions that are indexed by (ϖ, ψ) of the cluster center. On these bases, the discretization of HM paths can be conducted. The discretization procedures are described in Algorithm 2.

ALGORITHM 2: HM Path Discretization

Input: HM path, $\mathbf{V} = \{V_1, V_2 ...\}$
Output: The discrete sequence, $\mathbf{V}' = \{V_1', V_2' ...\}$

```
$Dis(V_1, V_2)$: calculates and returns the normalized orthodromic distance between $V_1$ and $V_2$.
$T(\varpi, \psi | V)$: shifts V according to action $(\varpi, \psi)$ and returns the result.
$a(\varpi, \psi | V)$: the active area with V as the center.
Sector allocation can be approximated by $i^* = \min_i Dis(T(\varpi_i, \psi_i | V'), V),\ i = 1, 2, 3 ... 48$

$V_t' = V_1$
$\mathbf{V}'$.push($V_t'$)
$V_{t+1} = \mathbf{V}$.pop()
while $V_{t+1}$ is not Null do
    if $V_{t+1} \in a(\varpi, \psi | V_t')$ then
        $i^* = \min_i Dis(T(\varpi_i, \psi_i | V_t'), V_{t+1}),\ i = 1, 2, 3 ... 48$
        $V_{t+1}' = T(\varpi_{i^*}, \psi_{i^*} | V_t')$
        $\mathbf{V}'$.push($V_{t+1}'$)
        $V_t' = V_{t+1}'$
        $V_{t+1} = \mathbf{V}$.pop()
    else
        while $V_{t+1} \notin a(\varpi, \psi | V_t')$ and $V_{t+1}$ is not Null do
            if $Dis(V_{t+1}, V_t') < \frac{15}{180} \cdot \pi$ then
                $V_{t+1} = \mathbf{V}$.pop()
                continue
            end if
            $i^* = \min_i Dis(T(\varpi_i, \psi_i | V_t'), V_{t+1}),\ i = 1, 2, 3 ... 48$
            $V_{t+1}' = T(\varpi_{i^*}, \psi_{i^*} | V_t')$
            $\mathbf{V}'$.push($V_{t+1}'$),
            $V_t' = V_{t+1}'$
        end while
    end if
end while
```

Dataset. We employ the publicly available Salient360 dataset [14] to conduct our experiments. There are 85 panoramas in the dataset, which can be classified into five categories: natural landscapes, grand halls, indoor rooms, cityscapes, and people. The resolutions of the images vary within the range from 5376 ×2688 to 18332 ×9166. During the experiment, 63 subjects were asked to watch the immersive image for 25 seconds through a VR display. In addition, there were 40 to 42 subjects participating in the evaluation for each image. Head orientations for each stimulus were provided. HM discretization is conducted on the dataset. We present some statistical results in Figure 5. From the results, we can observe that the transformation distances of most HMs are concentrated in a small range from 0.25 to 0.6 rad, and most transformation orientations are concentrated in the small range around the equator. As a result, the number of instances for different labels are imbalanced, which means that some classes dominate the other. To mitigate the class imbalance, new minority instances are synthesized by copying existing examples. In the test phase, besides the preceding dataset, we also test the performance on the OIQA dataset [26]. Sixteen raw images with resolution from 11332 × 5666 to 13320 × 6660 are included. Each image is evaluated by 20 subjects for 20 seconds. The ODS dataset [15] is also used to test the performance. The ODS dataset contains 22 images of resolution 8192 ×4096 that are viewed by 169 subjects. We mainly use the Salient360 dataset as the testbed. The OIQA and ODS datasets are used as a complement.

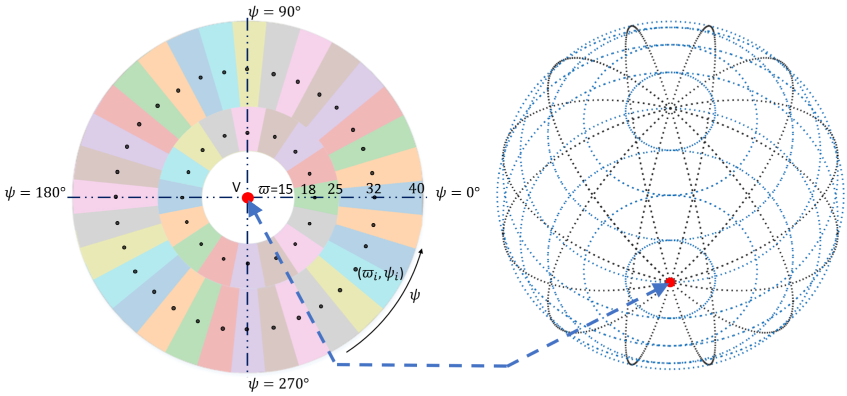


Fig. 4. There are 48 actions in multiple directions and distances. During the discretization, the pole is shifted to the viewport center, the transformation direction can be specified by the longitude of the meridian after the shift, and the transformation distance can be specified by the latitude after the shift that is also the radian between two positions on the sphere.

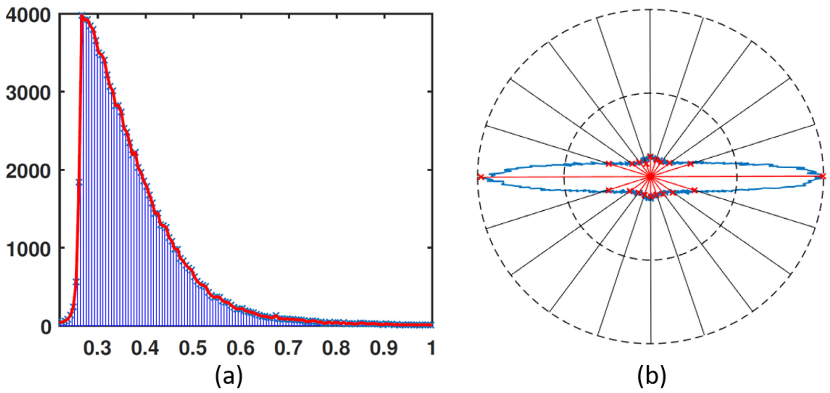


Fig. 5. (a) Histogram for the normalized orthodromic distance of HM transformations of which the unit is rad. (b) We count the HMs in different orientations and plot the distribution with a radar chart. The blue curve is the result after fitting.

Training and testing. The training of our model is conducted in two phases: SL and RL. To train and test our model, we randomly divide the Salient360 dataset into 80% training images and 20% testing images. There are 85 images in the dataset, of which 68 are used for training and 17 are used for testing. It is worth noting that to conduct the cross-dataset evaluation, all images in the OIQA and ODS datasets are not involved in the whole training process and are only used for testing.

In SL, we train the SL Q-learning network on the sampled state-action pairs, which are extracted from panoramas and the actions of human demonstrators in the training dataset. Cross entropy is employed as the loss function. The second stage of the training pipeline aims at improving the network by RL. Equation (11) indicates the loss function that is employed to update the network in the RL stage. The RL Q-learning network is identical in structure to the SL Q-learning network, and its weights are initialized to the same values. To improve stability, we use a separate network for generating the targets. The target network remains fixed between individual updates and is only updated every fixed step. In addition, the experience replay is used to remove the correlations of the input, and the ε-greedy policy is employed to select the action. We set the exploration rate ε as 1 at the beginning and anneal ε to a final value of 0.01. In the RL phase, the testing images in the Salient360 dataset are also involved. For these testing images in the Salient360 dataset, the exploration is conducted and stipulated by the psychovisual mechanism–based reward (R_f in Algorithm 1), which is independent of the annotation. To conduct cross-dataset evaluation, all images in the OIQA and ODS datasets are not involved in the SL phase and RL phase and are only used for testing. In the operation of HM prediction, the evaluation network will predict the action based on the current state. The new state will be fed to the network to make further prediction.

4.2 Evaluation Metric

To compare the distance between HM paths from the human and model, we employ to measure the dynamical similarity between HM paths of varied length. However, some existing methods totally ignore the temporal dimension [48]. By contrast, the modified metric takes both spatial properties and temporal dimension into account. In the modified metric, the similarity matrix is first computed, based on which the alignment is conducted and then the distances between the human HM path and the model-generated HM path are computed accordingly.

Similarity is measured by comparing both the normalized orthodromic distance and the angle distance of the direction. The calculations are performed in the three-dimensional Cartesian coordinate. Since image pixels are located on a sphere, the distance between two points is measured along the surface of the sphere. Thus, orthodromic distance is used instead of Euclidean distance. Given the HM location v_i from the human HM path and the HM location u_j from the model-generated HM path, the normalized orthodromic distance can be computed:

$$\mathrm{Dis}(v_i, u_j) = \arccos(v_i \cdot u_j). \tag{16}$$

Two sequentially adjacent HM locations form the vector $\overrightarrow{v_{i,i+1}}$ from the human and $\overrightarrow{u_{j,j+1}}$ from the model. Then the angle between two vectors can be calculated as the angle distance:

$$\mathrm{Ang}(\overrightarrow{v_{i,i+1}}, \overrightarrow{u_{j,j+1}}) = \arccos\left(\frac{\overrightarrow{v_{i,i+1}} \cdot \overrightarrow{u_{j,j+1}}}{\|\overrightarrow{v_{i,i+1}}\|_2 \|\overrightarrow{u_{j,j+1}}\|_2}\right). \tag{17}$$

Each HM location of the human path is compared with that of the model-generated path to form the similarity matrix W between two paths. The element of W is calculated as

$$w_{i,j} = \frac{1}{2\pi} * \mathrm{Dis}(v_i, u_j) + \frac{1}{2\pi} * \mathrm{Ang}(\overrightarrow{v_{i,i+1}}, \overrightarrow{u_{j,j+1}}). \tag{18}$$

The $w_{i,j}$ is normalized by the 2π to the 0 to 1 range. The alignment will be conducted based on the similarity matrix.

In the previous method for the path alignment [48], the alignment is conducted to reflect the dynamical similarities, but no temporal orders are considered. To ensure that the relative order is maintained during the alignment, the restriction is added that the alignment should be conducted forward. One sequence is arranged in rows, and another is arranged in columns that form the nodes of the graph. The edge weight is determined according to the similarity matrix indexed by the corresponding nodes. Figure 6 illustrates the placement of the nodes, connections, and edge weights. Dynamic programming can be employed to find the path of the smallest average orthodromic and angel distance (SA-OAD) from the start node (top left) to the goal node (bottom right). The alignment is global and spans the entire length of sequences, and the SA-OAD can be computed as

$$\text{SA-OAD} = \sum_{(i,j) \in P} w_{i,j} / L(P), \tag{19}$$

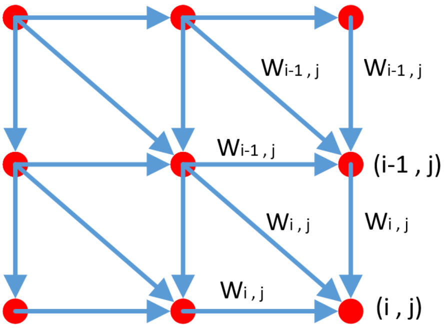


Fig. 6. Illustration of the placement for the nodes, connections, and edge weights during pairwise comparisons. One sequence is arranged in rows, and another is arranged in columns to form the nodes of the graph. The edge weight is determined according to the similarity matrix indexed by the corresponding nodes. Three directions are allowed to ensure that the alignment is conducted forward.

where P is the path from the starting node to the ending node determined by the alignment. The total weight of the path from the starting node is averaged by the path length $L(P)$.

In addition to the SA-OAD metric, we employ the inter-observer congruency (IOC) metric to measure the performance [49]. We implement the IOC measure by quantifying the saliency value at the locations of HMs [50]:

$$\mathrm{IOC} = \frac{1}{N}\sum_{i=1}^{N}\frac{I^s(V_i) - \mu_{I^s}}{\sigma_{I^s}}, \tag{20}$$

where V is the location of HMs, N is the total number of HMs, and I^s is the saliency map for HMs. μ and σ are the mean and standard deviation, respectively.

4.3 Experiments and Analyses

Our model leverages the observed HM locations to make the prediction. To evaluate accuracy, the initial k locations of each human HM path are taken as the observed locations, based on which our model will make future predictions.

Qualitative results. The performance of the proposed model has been explored in the qualitative perspective. Immersive images can be characterized by content complexity. The natural elements and artificial elements contribute to the content diversity. In addition, the presence of the human will also enrich the content. However, images can be characterized by the spatial complexity. Some images contain rich texture information, whereas others contain the plain areas of less texture information. As well, some images contain obvious foreground objects, whereas others have no obvious contrast between background and foreground. We choose 12 images of different characteristics: Gully, Crossroads1, Crossroads2, Highway1, Highway2, Hall, Yard, Train, Bar, Hotel, Room, and Square. The prediction of 15 HM locations are made on the 12 images. We divide the ground-truth HM paths into slices of 15 HM locations. Then we find the slice of minimum SA-OAD value from the observers' paths to the predicted path. The HM paths from predictions and the ground truth are plotted in the equirectangular image in Figure 7. From the results, we can observe that diversity predictions are made given different images, showing that the model can avoid the mode collapse. Although the left boundary of the image is 0 in longitude, and the right boundary is 2π in longitude, the agent can take the action that the starting position is near the left boundary and the target position is near the right boundary, which actually is a small movement on the sphere. This is because that the input of the model is the viewport data on the sphere that

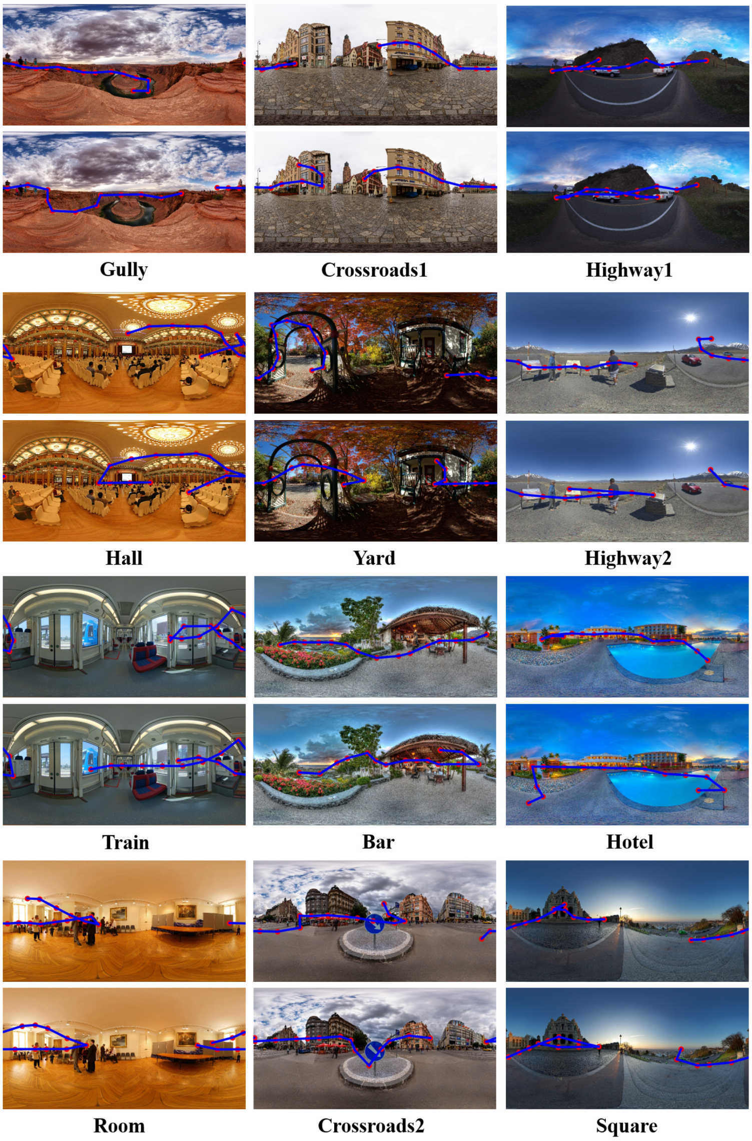


Fig. 7. Experimental results for HM path predictions under different panoramas. The prediction of 15 HM locations are made on the 12 images in the Salient360 dataset. The ground-truth HM paths are divided into slices of 15 HM locations. Then we find the slice of minimum SA-OAD value from the observers' paths to the predicted path. The HM paths from predictions and the ground truth are plotted in the equirectangular image. For each panorama, the first row is the ground-truth path and the second row is the prediction.

is a closed surface, and the actions that the agent can take are specified by the local transformation distance and orientation. The transformation distance is discretized into two levels, but they can tackle most conditions. Some longer transformations that rarely occur can be partitioned as combinations of the two levels of transformations. From Highway1, Highway2, and Room that contain obvious foreground objects, we can see that the predictions can cover and switch between the objects. By comparing the predictions and the ground truth, we can observe that the model

Table 3. Experimental Results for HM Predictions on the Salient360 [14], OIQA [26], and ODS [15] Datasets

Feature Type	Salient360		OIQA		ODS	
	SA-OAD ↓	IOC ↑	SA-OAD ↓	IOC ↑	SA-OAD ↓	IOC ↑
Random locations (RLS)	0.168	0.036	0.173	0.018	0.167	0.020
SHMP without DRL (SHMP$_b$, k=1)	0.131	1.378	0.142	1.309	0.136	1.321
Graph-based HM predictor (GBHMP) [10]	0.118	1.435	0.128	1.351	0.130	1.368
SHMP (SHMP$_e$, k=1)	0.102	1.581	0.124	1.413	0.122	1.422
SHMP (SHMP$_e$, k=6)	0.087	1.669	0.115	1.498	0.110	1.496

can perform well on the images of different characteristics. Because of RL and the discretization of HMs, our model is robust under different conditions.

Quantitative results. In the work of Zhu et al. [10], the graph-based HM path predictor employs both low-level and high-level features, which is compared as the baseline. In addition, paths of random HM locations are generated by randomly selecting locations on the sphere. Different from the baseline method, our proposed HM predictor predicts the HM in a stepwise way and is enhanced and generalized to more conditions by DRL. To compare the performance, the Salient360 [14], OIQA [26], and ODS [15] datasets are employed. We calculate the IOC value and the minimum SA-OAD value from the observers' paths to the predicted path. Table 3 compares the SA-OAD and IOC values of different methods. A lower SA-OAD value and a higher IOC value indicate that the prediction is more consistent with the ground-truth data. It can be observed that the proposed HM predictor without DRL can achieve good performance, and a substantial enhancement is made by DRL. The initial k locations of each human HM path are taken as the observed locations. The results demonstrate that more accurate predictions can be achieved with more observed information. The good performance is also maintained on the OIQA and ODS datasets, which demonstrates that the proposed method can also perform well in the cross-validation condition.

Longest path slice. The continuous head rotations in the horizontal directions contain rich information. We define the longest path slice as the longest continuous head rotations in the two horizontal directions, which should satisfy the condition that the longitude is monotonically increasing or decreasing. We calculate the Δ(longitude) of the longest path slice. The horizontal direction of the longest path slice is regarded as the positive direction, and the opposite direction is regarded as the negative direction. Table 4 shows the mean of Δ(longitude) of the continuous head rotation on the two directions. The value is normalized by 2π. It can be observed that the proposed method can well maintain the continuous head rotations in the horizontal directions. From the results, we can see that the rotation in the positive direction is generally greater than 0.5. The continuous rotations in the positive and negative directions cover a large portion of the longitude, and therefore much visual information is obtained during the exploration of continuous head rotations.

Multi-step predictions. To investigate the performance under increasing numbers of prediction steps, the HM predictor is manipulated to make N-step predictions. To compare the predictions, we divide the ground-truth HM paths into slices with the same length of N. Then we calculate the minimum SA-OAD value from the slices extracted from the observers' paths to the N-step predicted path. Figure 8 shows the minimum SA-OAD values on HM predictions with increasing numbers of prediction steps. It can be observed that the proposed predictor (SHMP$_e$) has the best performance on HM path prediction. When the number of prediction steps increases, the value of SA-OAD increases. There is a sharp decrease in the performance of the proposed HM predictor without DRL (SHMP$_b$) when the number of prediction steps increases. The results

Table 4. The Longest Path Slice Is Extracted for Each HM Path

Index	$\Delta(\mathrm{L})^p_G$	$\Delta(\mathrm{L})^n_G$	$\Delta(\mathrm{L})^p_P$	$\Delta(\mathrm{L})^n_P$
Gully	0.499	0.438	0.599	0.564
Crossroads1	0.604	0.464	0.711	0.517
Highway1	0.593	0.529	0.582	0.455
Hall	0.510	0.464	0.466	0.372
Yard	0.556	0.496	0.618	0.512
Highway2	0.589	0.444	0.649	0.466
Train	0.486	0.432	0.496	0.427
Bar	0.566	0.359	0.671	0.498
Hotel	0.622	0.491	0.624	0.453
Room	0.593	0.503	0.497	0.436
Crossroads2	0.520	0.473	0.633	0.573
Square	0.646	0.459	0.690	0.515

The Δ(longitude) of the longest path slice is calculated and averaged on each panorama. Here, *p* refers to positive, *n* refers to negative, *G* refers to ground truth, and*P* refers to predicted. SHMP$_e$, k=6 is employed to make the predictions.

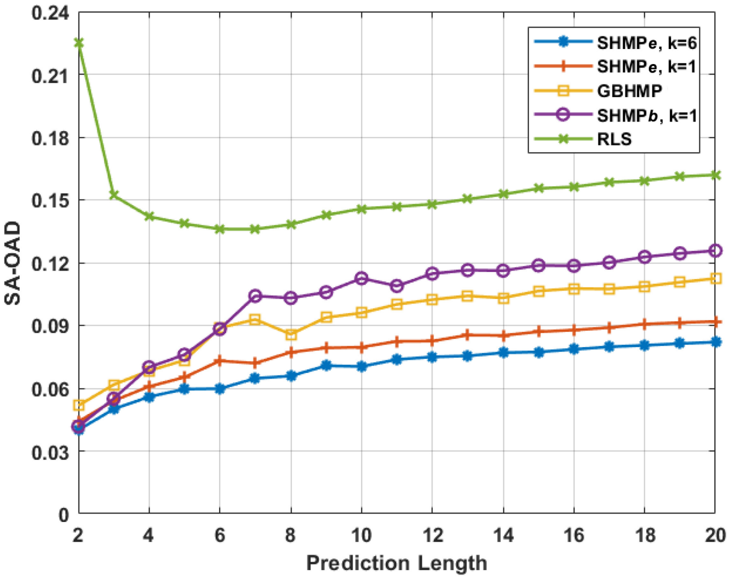
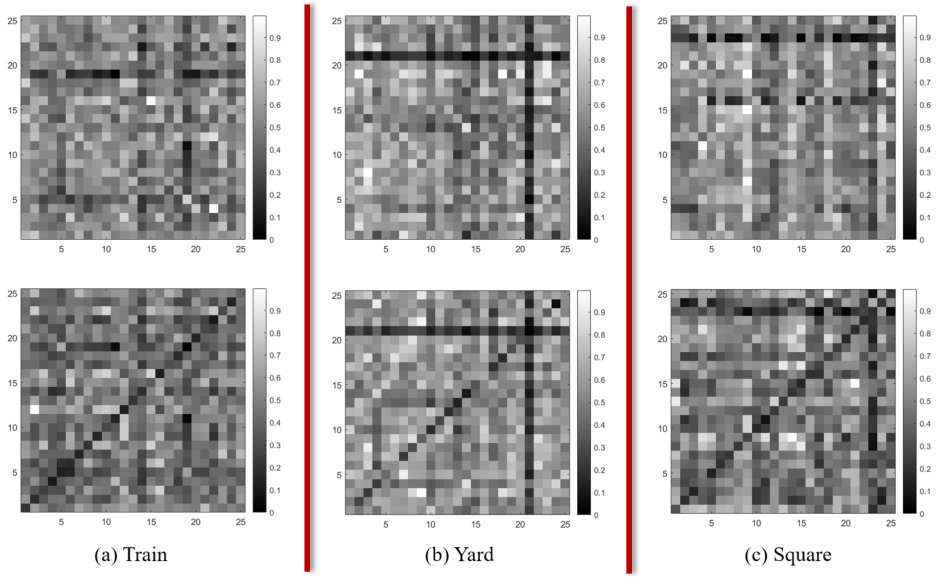


Fig. 8. The HM predictor is manipulated to make multi-step predictions on the testing Salient360 dataset. The ground-truth HM paths are divided into slices with the same length. The minimum SA-OAD value is calculated from the slices extracted from the observers' paths to the predicted path. The minimum SA-OAD value is reported.

demonstrate that the propagation of discrepancy makes the predicted HM paths deviate from the visual behavior pattern of viewers as the length of the sequence increases. The employment of psychovisual mechanism–based reward can guide the agent's action during the exploration to enhance the robustness, and more observed information can make the prediction more accurate.

Inter-paths comparisons. Given more observations of the HM path, the predicted HM path should be more consistent with the observed path from the demonstrator. Therefore, we also provide the



(a) Train (b) Yard (c) Square

Fig. 9. Similarity matrices are plotted to make the inter-paths comparison. The horizontal axis represents predicted HM paths, and the vertical axis represents ground-truth HM paths. Predicted paths are arranged in the same order with the corresponding observed paths. The first row shows the results of SHMP$_e$, k=1, and the second row shows the results of SHMP$_e$, k=8.

inter-paths comparisons. To measure the consistency between the predicted HM paths and the ground-truth HM paths, we provide the similarity matrix of SA-OAD values from HM paths from demonstrators to predicted HM paths. Predicted paths are arranged in the same order with the corresponding observed paths (i.e., there are 25 pairs). The horizontal axis represents predicted HM paths, and the vertical axis represents ground-truth HM paths. Min-max normalization is used to compare the results in the unit range. As shown in Figure 9, the proposed model predicts the HM paths on three panoramas: Train, Yard, and Square. The model is manipulated to conduct the predictions under different amounts of available observations. It can be observed from the diagonal of the similarity matrix that when more observations are available, the predicted HM path is more consistent with the observed path from the demonstrator.

5 CONCLUSION AND FUTURE WORK

In this article, we focus on the prediction of HMs, which are important behaviors of viewers, and propose the effective SHMP for panoramas that can be used to increase the bandwidth efficiency in the video streaming, increase the visual quality, and reduce the motion-to-photon delay. We employ the framework of DRL to predict HM in immersive images. Our model extracts the features of the viewport images. The spherical coordinates of each pixel are provided as extra position information to enable the model to observe the position prior. To enable the model to learn the impact of past HMs, the visited map recording the historical HM positions is also included in the input. In the training pipeline of the designed network, on one hand the training is designed such that it includes the employment of available annotation data to reflect the closeness of the truly annotation data to the predictions of our model. On the other hand, the psychovisual mechanism–based reward is independent of the annotation and is used to maximize the exploration and enable the model to generalize to more conditions. The experimental results on the publicly available dataset demonstrate the effectiveness of our method. In the future, we are ready to work on improving the framework of our model and adapting our model for HM predictions on omnidirectional videos. One intuitive idea is to incorporate the temporal information.

REFERENCES

[1] TOBII VR. 2019. Discover New Possibilities with Eye Tracking in VR. Retrieved September 23, 2020 from https://vr.tobii.com/.

[2] Afshin Taghavi Nasrabadi, Anahita Mahzari, Joseph D. Beshay, and Ravi Prakash. 2017. Adaptive 360-degree video streaming using scalable video coding. In *Proceedings of the 25th ACM International Conference on Multimedia*. ACM, New York, NY, 1689–1697.

[3] J. M. P. Van Waveren. 2016. The asynchronous time warp for virtual reality on consumer hardware. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. ACM, New York, NY, 37–46.

[4] Ana De Abreu, Cagri Ozcinar, and Aljosa Smolic. 2017. Look around you: Saliency maps for omnidirectional images in VR applications. In *Proceedings of the 9th International Conference on Quality of Multimedia Experience*. IEEE, Los Alamitos, CA, 1–6.

[5] Yashas Rai, Patrick Le Callet, and Philippe Guillotel. 2017. Which saliency weighting for omni directional image quality assessment? In *Proceedings of the International Conference on Quality of Multimedia Experience*. IEEE, Los Alamitos, CA, 1–6.

[6] Mikhail Startsev and Michael Dorr. 2018. 360-Aware saliency estimation with conventional image saliency predictors. *Signal Processing: Image Communication* 69 (2018), 43–52.

[7] Federica Battisti, Sara Baldoni, Michele Brizzi, and Marco Carli. 2018. A feature-based approach for saliency estimation of omni-directional images. *Signal Processing: Image Communication* 69 (2018), 53–59.

[8] Jing Ling, Kao Zhang, Yingxue Zhang, Daiqin Yang, and Zhenzhong Chen. 2018. A saliency prediction model on 360 degree images using color dictionary based sparse representation. *Signal Processing: Image Communication* 69 (2018), 60–68.

[9] Pierre Lebreton and Alexander Raake. 2018. GBVS360, BMS360, ProSal: Extending existing saliency prediction models from 2D to omnidirectional images. *Signal Processing: Image Communication* 69 (2018), 69–78.

[10] Yucheng Zhu, Guangtao Zhai, and Xiongkuo Min. 2018. The prediction of head and eye movement for 360 degree images. *Signal Processing: Image Communication* 69 (2018), 15–25.

[11] Rafael Monroy, Sebastian Lutz, Tejo Chalasani, and Aljosa Smolic. 2018. SalNet360: Saliency maps for omni-directional images with CNN. *Signal Processing: Image Communication* 69 (2018), 26–34.

[12] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. 2018. Cube padding for weakly-supervised saliency prediction in 360 videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 1420–1429.

[13] Yanyu Xu, Yanbing Dong, Junru Wu, Zhengzhong Sun, Zhiru Shi, Jingyi Yu, and Shenghua Gao. 2018. Gaze prediction in dynamic 360° immersive videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 5333–5342.

[14] Yashas Rai, Jesús Gutiérrez, and Patrick Le Callet. 2017. A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM Multimedia Systems Conference*. ACM, New York, NY, 205–210.

[15] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. 2018. Saliency in VR: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1633–1642.

[16] Xavier Corbillon, Francesca De Simone, and Gwendal Simon. 2017. 360-Degree video head movement dataset. In *Proceedings of the 8th ACM Multimedia Systems Conference*. ACM, New York, NY, 199–204.

[17] Benjamin J. Li, Jeremy N. Bailenson, Adam Pines, Walter J. Greenleaf, and Leanne M. Williams. 2017. A public database of immersive VR videos with corresponding ratings of arousal, valence, and correlations between head movements and self report measures. *Frontiers in Psychology* 8 (2017), 2116.

[18] Stephan Fremerey, Ashutosh Singla, Kay Meseberg, and Alexander Raake. 2018. AVtrack360: An open dataset and software recording people's head rotations watching 360° videos on an HMD. In *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, New York, NY, 403–408.

[19] Erwan J. David, Jesús Gutiérrez, Antoine Coutrot, Matthieu Perreira Da Silva, and Patrick Le Callet. 2018. A dataset of head and eye movements for 360 videos. In *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, New York, NY, 432–437.

[20] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*. 1928–1937.

[21] Giacomo Rizzolatti, Lucia Riggio, Isabella Dascola, and Carlo Umilta. 1987. Reorienting attention across the horizontal and vertical meridians: Evidence in favor of a premotor theory of attention. *Neuropsychologia* 25, 1A (1987), 31–40.

[22] Pierre-Yves Oudeyer and Frederic Kaplan. 2007. What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics* 1 (2007), 6.

[23] Guangtao Zhai, Xiaolin Wu, Xiaokang Yang, Weisi Lin, and Wenjun Zhang. 2012. A psychovisual quality metric in free-energy principle. *IEEE Transactions on Image Processing* 21, 1 (2012), 41–52.

[24] Laurent Itti and Pierre F. Baldi. 2006. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems*. 547–554.

[25] Karl Friston. 2010. The free-energy principle: A unified brain theory?*Nature Reviews Neuroscience* 11, 2 (2010), 127.

[26] Huiyu Duan, Guangtao Zhai, Xiongkuo Min, Yucheng Zhu, Yi Fang, and Xiaokang Yang. 2018. Perceptual quality assessment of omnidirectional images. In *Proceedings of the IEEE International Symposium on Circuits and Systems*. IEEE, Los Alamitos, CA, 1–5.

[27] Cagri Ozcinar and Aljosa Smolic. 2018. Visual attention in omnidirectional video for virtual reality applications. In *Proceedings of the 10th International Conference on Quality of Multimedia Experience*. IEEE, Los Alamitos, CA, 1–6.

[28] Hou Ning Hu, Yen Chen Lin, Ming Yu Liu, Hsien Tzu Cheng, Yung Ju Chang, and Min Sun. 2017. Deep 360 pilot: Learning a deep agent for piloting through 360 sports video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 1396–1405.

[29] Wei-Sheng Lai, Yujia Huang, Neel Joshi, Christopher Buehler, Ming-Hsuan Yang, and Sing Bing Kang. 2018. Semantic-driven generation of hyperlapse from 360 degree video. *IEEE Transactions on Visualization and Computer Graphics* 24, 9 (2018), 2610–2621.

[30] Yu-Chuan Su and Kristen Grauman. 2017. Learning spherical convolution for fast features from 360° imagery. In *Advances in Neural Information Processing Systems*. 529–539.

[31] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. 2018. Spherical CNNs. arXiv:1801.10130

[32] George A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information.*Psychological Review* 63, 2 (1956), 81.

[33] Holger Kantz and Thomas Schreiber. 2004. *Nonlinear Time Series Analysis*. Cambridge University Press.

[34] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[35] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 770–778.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[38] Benjamin W. Tatler. 2007. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision* 7, 14 (11 2007), 4.

[39] Evgeniy Upenik and Touradj Ebrahimi. 2017. A simple method to obtain visual attention data in head mounted virtual reality. In *Proceedings of the IEEE International Conference on Multimedia Expo Workshops*. IEEE, Los Alamitos, CA, 73–78.

[40] Ruzena Bajcsy. 1988. Active perception. *Proceedings of the IEEE* 76, 8 (1988), 966–1005.

[41] Ke Gu, Guangtao Zhai, Weisi Lin, Xiaokang Yang, and Wenjun Zhang. 2015. Visual saliency detection with free energy theory. *IEEE Signal Processing Letters* 22, 10 (2015), 1552–1555.

[42] Jinjian Wu, Guangming Shi, Weisi Lin, Anmin Liu, and Fei Qi. 2013. Just noticeable difference estimation for images with free-energy principle. *IEEE Transactions on Multimedia* 15, 7 (2013), 1705–1710.

[43] Thomas Schmidt and Dirk Vorberg. 2006. Criteria for unconscious cognition: Three types of dissociation. *Perception & Psychophysics* 68, 3 (2006), 489–504.

[44] Hagai Attias. 2000. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems*. 209–215.

[45] Stephen J. Guastello. 2013. *Human Factors Engineering and Ergonomics: A Systems Approach*. CRC Press, Boca Raton, FL.

[46] Neeraj J. Gandhi, Ellen J. Barton, and David L. Sparks. 2008. Coordination of eye and head components of movements evoked by stimulation of the paramedian pontine reticular formation. *Experimental Brain Research* 189, 1 (2008), 35.

[47] Michael F. Goodchild. 2007. Citizens as sensors: The world of volunteered geography. *GeoJournal* 69, 4 (2007), 211–221.

[48] Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, and Yuan Yao. 2011. Simulating human saccadic scanpaths on natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 441–448.

[49] Olivier Le Meur and Thierry Baccino.2013. Methods for comparing scanpaths and saliency maps: Strengths and weaknesses. *Behavior Research Methods* 45, 1 (2013), 251–266.

[50] Robert J. Peters, Asha Iyer, Laurent Itti, and Christof Koch. 2005. Components of bottom-up gaze allocation in natural images. *Vision Research* 45, 18 (2005), 2397–2416.

Received January 2020; revised June 2020; accepted July 2020